



卓越工程师培养计划

▪ EDA ▪

<http://www.phei.com.cn>

周润景 刘晓霞 编著



基于 PROTEUS

的电路设计、 仿真与制板



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY



卓越工程师培养计划

▪ EDA ▪

<http://www.phei.com.cn>

周润景 刘晓霞 编著



基于 PROTEUS

的电路设计、 仿真与制板

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书基于 PROTEUS 8.0 SP0 版本软件, 以软件的实际操作过程为写作的次序, 以丰富的实例贯穿全书进行全面的讲解, 包括 PROTEUS 软件的操作、模拟和数字电路的分析方法、单片机电路的软硬件调试技术以及 PCB 设计方法等。

本书面向实际、图文并茂、内容详细具体、通俗易懂、层次分明、易于掌握, 可以为电子产品研发、电路系统教学, 以及课程设计、毕业设计和电子设计竞赛等提供很大的帮助。

本书既可以作为从事电子设计的工程技术人员自学的参考书, 也可以作为高等院校相关专业的教材或职业培训教材。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

基于 PROTEUS 的电路设计、仿真与制板 / 周润景, 刘晓霞编著. —北京: 电子工业出版社, 2013.8
(卓越工程师培养计划)

ISBN 978-7-121-21090-7

I. ①基… II. ①周… ②刘… III. ①单片微型计算机—系统仿真—应用软件 IV. ①TP368.1

中国版本图书馆 CIP 数据核字 (2013) 第 171087 号

策划编辑: 张 剑 (zhang@phei.com.cn)

责任编辑: 刘真平

印 刷:

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 16.5 字数: 422.4 千字

印 次: 2013 年 8 月第 1 次印刷

印 数: 4 000 册 定价: 45.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

随着电子技术的飞速发展，电子设计的方式也不断进步。PROTEUS 虚拟开发仿真平台是一款可以实现数字电路、模拟电路、微控制器系统仿真以及 PCB 设计等功能的 EDA 软件。电路的软、硬件的设计与调试都是在计算机虚拟环境下进行的。基于这一设计思想开发的 PROTEUS 软件，可以在原理图设计阶段对所设计的电路进行验证，并可以通过改变元件参数使整个电路性能达到最优化。这样就避免了传统电子电路设计中方案更换带来的多次重复购买元件及制板的麻烦，可以节省很多时间和经费，也提高了设计的效率和质量。

PROTEUS 软件集强大的功能与简易的操作于一体，成为嵌入式系统领域技术最先进的开发工具。PROTEUS 软件提供了 30 多个元件库、上万个元件。元件涉及电阻、电容、二极管、晶体管、MOS 管、变压器、继电器、各种放大器、各种激励源、900 多种微控制器、各种逻辑电路和各种仪表等。在 PROTEUS 软件中提供的仪表有交直流电压表、交直流电流表、逻辑分析仪、定时/计数器、液晶屏、LED、按钮、键盘等外设，同时支持图形化的分析功能，具有直流工作点、瞬态特性、交直流参数扫描、频率特性、傅里叶分析、失真分析、噪声分析等多种分析功能，并可将仿真曲线绘制到图表中。

本书基于 PROTEUS 8.0 SP0 版本的软件，通过实例讲解 PROTEUS 软件的操作，包括原理图输入、电路仿真、软件调试及系统协同仿真等。

本书总共分为 8 章，其主要内容如下：

第 1 章：PROTEUS 原理图编辑环境及 PROTEUS ISIS 软件的菜单栏、工具栏及编辑窗口导航。

第 2 章：介绍原理图的设计方法及步骤，包括查找、放置元件，原理图连线等。

第 3 章：介绍电路分析的工具，包括激励源、仿真图表和虚拟仪器的使用方法。

第 4 章：介绍单片机的设计与仿真方法，包括源代码的编辑、目标代码的生成、第三方编辑器和第三方 IDE 的使用、单片机系统的调试及系统仿真。

第 5 章：介绍利用 PROTEUS 软件进行仿真的多个例子，包括异步十进制计数器、RS 触发器、竞赛抢答器等。

第 6 章：介绍音频放大器的设计，主要包括直流电源、放大电路、功率放大器的设计及分析等。

第 7 章：介绍单片机的设计实例，主要是信号发生器的硬件和软件设计及实际仿真。

第 8 章：介绍 PCB 的设计，主要包括原理图的后处理、创建元件封装、PCB 布局、PCB 布线以及光绘文件的输出。

本书由周润景、刘晓霞编著。其中刘晓霞编写了第 8 章，其余章节由周润景编写。张丽娜、张红敏、张丽敏、贾雯、托亚、姜攀、张龙龙、张鹏飞、任冠中、丁莉、王志军、胡训智、李琳参与了本书的编写。全书由周润景教授统稿。

由于作者水平有限，加上时间仓促，不妥之处敬请广大读者批评指正。

编著者

目 录

第 1 章	PROTEUS 概述	1
1.1	PROTEUS ISIS 及 ARES 概述	1
1.2	PROTEUS ISIS 编辑环境	2
1.3	PROTEUS ISIS 菜单栏介绍	5
1.4	编辑窗口显示导航	8
1.5	编辑窗口的设置	9
第 2 章	PROTEUS ISIS 原理图设计	16
2.1	PROTEUS ISIS 原理图输入流程	16
2.2	原理图设计方法与步骤	17
2.3	PROTEUS ISIS 编辑窗口连接端子	27
第 3 章	电路分析	30
3.1	PROTEUS ISIS 激励源	30
3.1.1	直流信号源 (DC Generator)	30
3.1.2	正弦波信号源 (SINE Generator)	32
3.1.3	模拟脉冲信号源 (PULSE Generator)	34
3.1.4	指数脉冲信号源 (EXP Generator)	36
3.1.5	单频率调频波信号源 (SFFM Generator)	38
3.1.6	分段线性激励源 (PWLIN Generator)	39
3.1.7	FILE 信号源 (FILE Generator)	40
3.1.8	音频信号源 (AUDIO Generator)	42
3.1.9	数字单稳态逻辑电平信号源 (DSTATE Generator)	43
3.1.10	数字单边沿信号源 (DEDGE Generator)	45
3.1.11	单周期数字脉冲信号源 (DPULSE Generator)	46
3.1.12	数字时钟信号源 (DCLOCK Generator)	47
3.1.13	数字模式信号源 (DPATTERN Generator)	48
3.2	基于图表的分析	50
3.2.1	基于模拟分析图表的电路分析	51
3.2.2	基于数字分析图表的电路分析	54
3.2.3	基于混合分析图表的电路分析	55
3.2.4	基于频率分析图表的电路分析	57
3.2.5	基于转移特性分析图表的电路分析	59
3.2.6	基于噪声分析图表的电路分析	60
3.2.7	基于失真分析图表的电路分析	62
3.2.8	基于傅里叶分析图表的电路分析	65

基于 PROTEUS 的电路设计、仿真与制板

3.2.9	基于音频分析图表的电路分析	66
3.2.10	基于交互分析图表的电路分析	68
3.2.11	基于一致性分析图表的电路分析	70
3.2.12	基于直流扫描分析图表的电路分析	73
3.2.13	基于交流扫描分析图表的电路分析	74
3.3	虚拟仪器	76
3.3.1	虚拟示波器 (Oscilloscope)	76
3.3.2	逻辑分析仪 (Logic Analyser)	79
3.3.3	计数器/定时器 (Counter Timer)	80
3.3.4	虚拟终端 (Virtual Terminal)	82
3.3.5	SPI 调试器 (SPI Debugger)	85
3.3.6	I ² C 调试器 (I ² C Debugger)	87
3.3.7	信号发生器 (Signal Generator)	89
3.3.8	模式发生器 (Pattern Generator)	90
3.3.9	电压表和电流表 (AC/DC Voltmeter/Ammeter)	97
3.4	探针	98
3.4.1	电压探针	98
3.4.2	电流探针	99
第 4 章	PROTEUS ISIS 单片机仿真	100
4.1	在 PROTEUS ISIS 中输入单片机系统电路	100
4.1.1	选取仿真元件	100
4.1.2	调试 PWM 输出电路中的 ADC0808 模数转换电路	101
4.1.3	设计 PWM 输出控制电路	104
4.2	在 PROTEUS ISIS 中进行软件编程	106
4.2.1	在 PROTEUS ISIS 中创建源代码文件	107
4.2.2	在 PROTEUS ISIS 中将源代码文件生成目标代码	109
4.3	进行系统调试	110
4.3.1	将目标代码添加到电路	110
4.3.2	进行电路调试	112
4.3.3	仿真电路	124
4.4	将 PROTEUS 与 Keil 联调	128
4.4.1	使用 Keil 的 μ Vision3 集成开发环境	129
4.4.2	进行 PROTEUS 与 Keil 的整合	142
4.4.3	进行 PROTEUS 与 Keil 的联调	143
第 5 章	基于 PROTEUS ISIS 的数字电路分析	150
5.1	异步十进制计数器电路及其分析	150
5.1.1	异步十进制计数器电路	150
5.1.2	异步十进制计数器电路分析	153
5.1.3	异步十进制计数器电路完善	155
5.2	RS 触发器电路分析——数字模式信号源与数字分析	157

目 录

5.2.1	RS 触发器电路	157
5.2.2	RS 触发器电路调试	159
5.2.3	RS 触发器数字图表分析——数字模式信号源编辑	160
5.2.4	RS 触发器数字图表分析——探针及数字分析图表编辑	161
5.2.5	RS 触发器电路分析	162
5.2.6	RS 触发器用于消除机械开关振荡引起的脉冲	164
5.3	竞赛抢答器电路分析——数字单周期脉冲信号源与数字分析	166
5.3.1	竞赛抢答器电路	167
5.3.2	数字时钟信号源及数字单周期脉冲信号源编辑	169
5.3.3	竞赛抢答器电路分析	169
5.3.4	利用灌电流和或非门设计竞赛抢答器电路	173
第 6 章	模拟电路设计实例——音频功率放大器的设计	177
6.1	设计任务及要求	177
6.2	音频功率放大器简介	177
6.3	直流稳压源设计	178
6.3.1	原理分析与设计	178
6.3.2	计算机仿真分析	178
6.4	音调控制电路	182
6.4.1	原理分析与设计	183
6.4.2	计算机辅助设计与分析	185
6.5	工频陷波器	189
6.5.1	原理分析与设计	189
6.5.2	计算机仿真分析	191
6.6	前级放大电路	195
6.6.1	原理分析与设计	196
6.6.2	计算机仿真分析	196
6.7	功率放大电路	197
6.7.1	原理分析与设计	197
6.7.2	计算机仿真分析	199
6.8	电路整体的协调及仿真	201
6.8.1	电路各组成部分的协调连接	201
6.8.2	带通滤波器的加入	202
6.8.3	计算机辅助设计与分析	203
6.8.4	电路整体的计算机仿真分析与验证	205
第 7 章	单片机设计实例——信号发生器的设计	208
7.1	设计任务及要求	208
7.2	原理图设计	208
7.3	汇编语言程序设计流程	212
7.4	汇编语言程序源代码	213
7.5	C 语言程序源代码	216

7.6	系统仿真	219
第 8 章	PROTEUS ARES PCB 设计	221
8.1	原理图的后处理	221
8.1.1	概述	221
8.1.2	自定义元件符号	222
8.1.3	检查元件的封装属性	231
8.1.4	完善原理图	233
8.2	创建元件封装	233
8.2.1	元件符号与元件封装	233
8.2.2	创建元件封装	236
8.2.3	指定元件封装	239
8.3	PCB 布局	242
8.3.1	设置层面	242
8.3.2	自动布局	242
8.3.3	手工布局	244
8.3.4	调整文字	245
8.4	PCB 布线	246
8.4.1	设置约束规则	246
8.4.2	手工布线及自动布线	248
8.4.3	铺铜	252
8.5	输出光绘文件	254
	参考文献	256

第1章 PROTEUS 概述

PROTEUS 软件是由英国 Labcenter Electronics 公司开发的 EDA 工具软件，由 ISIS 和 ARES 两个软件构成，其中 ISIS 是一款便捷的电子系统仿真平台软件，ARES 是一款高级的布线编辑软件，它集成了高级原理布图、混合模式 SPICE 电路仿真、PCB 设计以及自动布线来实现一个完整的电子设计。



1.1 PROTEUS ISIS 及 ARES 概述

1. PROTEUS ISIS 概述

通过 PROTEUS ISIS 软件的 VSM（虚拟仿真技术），用户可以对模拟电路、数字电路、模数混合电路，以及基于微控制器的系统连同所有外围接口电子器件一起仿真。

PROTEUS VSM 有两种截然不同的仿真方式：交互式仿真和基于图表的仿真。其中交互式仿真可实时观测电路的输出，因此可用于检验设计的电路是否能正常工作；而基于图表的仿真能够在仿真过程中放大一些特别的部分，进行一些细节上的分析，因此基于图表的仿真可用于研究电路的工作状态和进行细节的测量。

PROTEUS 软件的模拟仿真直接兼容厂商的 SPICE 模型，采用扩充了的 SPICE3F5 电路仿真模型，能够记录基于图表的频率特性、直流电的传输特性、参数的扫描、噪声的分析、傅里叶分析等，具有超过 8 000 种的电路仿真模型。

PROTEUS 软件的数字仿真支持 JDEC 文件的物理器件仿真，有全系列的 TTL 和 CMOS 数字电路仿真模型，同时一致性分析易于系统的自动测试。PROTEUS 软件支持许多通用的微控制器，如 PIC、AVR、HC11 及 8 051；包含强大的调试工具，可对寄存器、存储器实时监测；具有断点调试功能及单步调试功能；可对显示器、按钮、键盘等外设进行交互可视化仿真。此外，PROTEUS 可对 IAR C-SPY、Keil μ Vision3 等开发工具的源程序进行调试，可与 Keil 实现联调。

2. PROTEUS ARES 概述

PROTEUS ARES PCB 的设计采用了原 32 位数据库的高性能 PCB 设计系统，以及高性能的自动布局和自动布线算法；支持多达 16 个布线层、2 个丝网印刷层、4 个机械层，加上线路板边界层、布线禁止层、阻焊层，可以在任意角度放置元件和焊盘连线；支持光绘文件的生成；具有自动的门交换功能；集成了高度智能的布线算法；有超过 1 000 个标准的元件引脚封装；支持输出各种 Windows 设备；可以导出其他线路板设计工具的文件格式；能自动插入最近打开的文档；元件可以自动放置。



1.2 PROTEUS ISIS 编辑环境

PROTEUS 电路设计是在 PROTEUS ISIS 环境中进行绘制的。PROTEUS ISIS 编辑环境具有友好的人机交互界面，而且设计功能强大，使用方便，易于上手。

PROTEUS ISIS 可运行于 Windows PE/2000/XP/7 及更高操作系统，其对 PC 的配置要求不高，一般的配置就能满足要求。

单击“开始”菜单，选择“Proteus 8 Professional”程序，在出现的子菜单中选择“Proteus 8 Professional”选项，如图 1-1 所示。

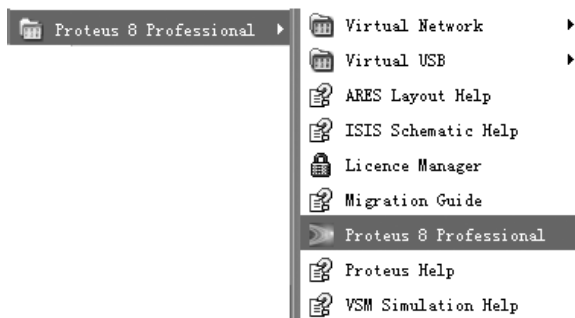


图 1-1 选择“Proteus 8 Professional”选项

系统启动界面如图 1-2 所示。



图 1-2 PROTEUS ISIS 启动界面

创建新项目，如图 1-3 所示。



图 1-3 PROTEUS ISIS 创建新项目

在向导的第一页指定一个项目名称和保存路径，如图 1-4 所示。

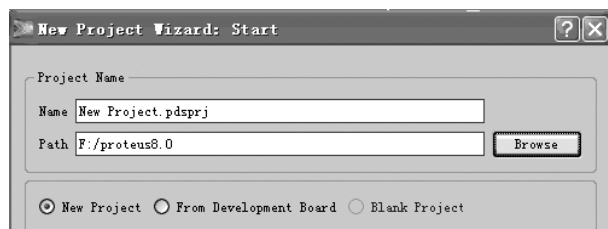


图 1-4 指定项目名称和保存路径

选择默认的原理图设计模板，如图 1-5 所示。

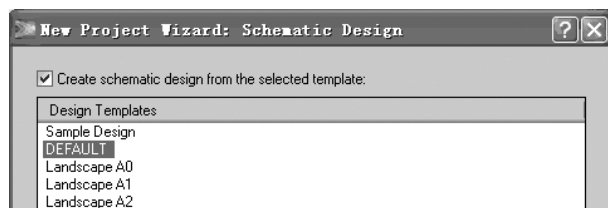


图 1-5 创建原理图设计模板

单击“Next”按钮，同样的，我们需要一个便于检查的 PCB 的布局页面，选择默认的模板，如图 1-6 所示。

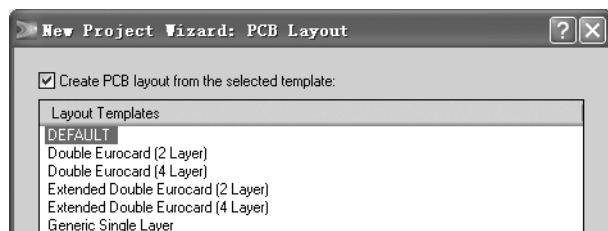


图 1-6 创建 PCB 布局模板

单击“Next”按钮，出现创建固件的界面，如图 1-7 所示。



图 1-7 创建固件

因为我们不是模拟设计，所以离开创建固件页面，继续单击“Next”按钮，出现创建项目的简要说明，如图 1-8 所示。

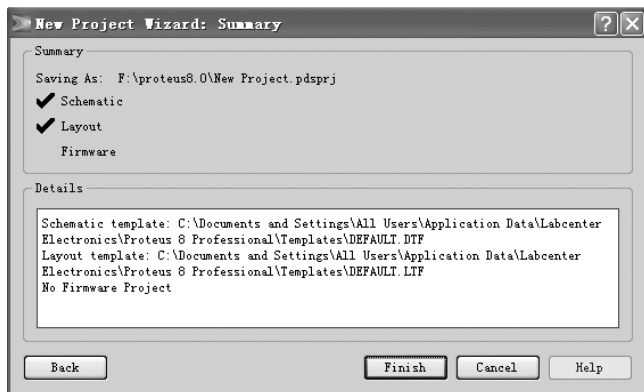


图 1-8 项目说明

单击“Finish”按钮来完成项目创建。该项目由两个选项卡打开，一个是原理图编辑，另一个用于 PCB 布局。单击 ISIS 选项卡，将原理图编辑置于前台，如图 1-9 所示。

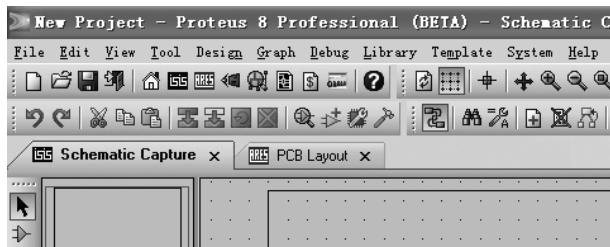


图 1-9 原理图设计选项卡

之后系统进入 PROTEUS ISIS 编辑环境，如图 1-10 所示。

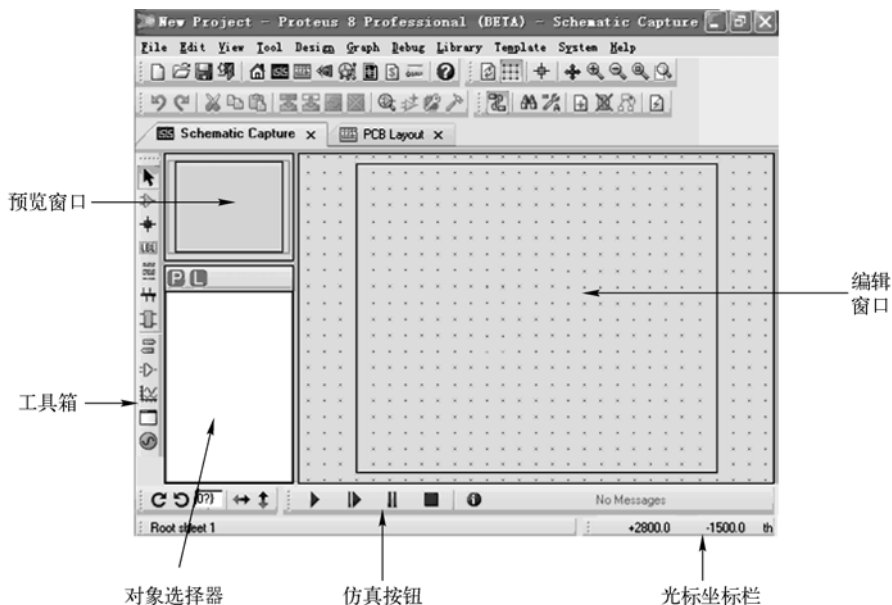


图 1-10 PROTEUS ISIS 编辑环境

编辑窗口用于放置元件，进行连线，绘制原理图。预览窗口通常用来显示全部原理图。蓝框表示当前页的边界，绿框表示当前编辑窗口显示的区域。但当从对象选择器中选一个新的对象时，预览窗口将显示选中的对象。



1.3 PROTEUS ISIS 菜单栏介绍

1. 主菜单

PROTEUS ISIS 的主菜单栏包括 File（文件）、Edit（编辑）、View（视图）、Tool（工具）、Design（设计）、Graph（图形）、Debug（调试）、Library（库）、Template（模板）、System（系统）、Help（帮助），如图 1-11 所示。



图 1-11 PROTEUS ISIS 的主菜单和主工具栏

- ☺ File 菜单：包括新建设计、打开设计、保存设计、导入/导出文件，也可以用于打印、显示设计文档，以及退出 PROTEUS ISIS 系统等。
- ☺ Edit 菜单：包括撤销/恢复操作，查找与编辑元件，剪切、复制、粘贴对象，以及设置多个对象的层叠关系等。
- ☺ View 菜单：包括是否显示网格、设置格点间距、缩放电路图及显示与隐藏各种工具栏等。
- ☺ Tool 菜单：工具菜单。它包括实时注解、自动布线、查找并标记、属性分配工具、全局注解、导入文本数据、元件清单、电气规则检查、编译网络标号、编译模型、将网络标号导入 PCB，以及从 PCB 返回原理图设计等工具栏。
- ☺ Design 菜单：工程设计菜单。它具有编辑设计属性，编辑原理图属性，编辑设计说明，配置电源，新建、删除原理图，在层次原理图中总图与子图以及各子图之间相互跳转和设计目录管理等功能。
- ☺ Graph 菜单：图形菜单。它具有编辑仿真图形，添加仿真曲线、仿真图形，查看日志，导出数据，清除数据和一致性分析等功能。
- ☺ Debug 菜单：调试菜单。包括启动调试、执行仿真、单步运行、断点设置和重新排布弹出窗口等功能。
- ☺ Library 菜单：库操作菜单。它具有选择元件及符号、制作元件及符号、设置封装工具、分解元件、编译库、自动放置库、校验封装和调用库管理器等功能。
- ☺ Template 菜单：模板菜单。包括设置图形格式、文本格式、设计颜色以及连接点和图形等。
- ☺ System 菜单：系统设置菜单。包括设置系统环境、路径、图纸尺寸、标注字体、热键以及仿真参数和模式等。

☺ Help 菜单：帮助菜单。包括版权信息、PROTEUS ISIS 学习教程和示例等。

2. 主工具栏

PROTEUS ISIS 的主工具栏位于主菜单下面两行，以图标形式给出，包括 File 工具栏、View 工具栏、Edit 工具栏和 Design 工具栏四个部分。工具栏中每一个按钮都对应一个具体的菜单命令，便于快捷地使用命令。主工具栏按钮功能如表 1-1 所示。



















表 1-1 主工具栏按钮功能

按 钮	对 应 菜 单	功 能
	Home Page	打开主页
	Schematic Capture	原理图输入
	PCB Layout	PCB 布局
	3D Visualizer	3D 观察器
	Gerber Viewer	PCB 观察器
	Bill of Materials	材料清单
	Source Code	源文件菜单
	Design Explorer	设计资源管理器
	File→New Project	新建项目
	File→Open Project	打开项目
	File→Save Project	保存项目
	File→Close Project	关闭项目
	View→Redraw Display	刷新
	View→Toggle Grid	栅格开关
	View→Toggle False Origin	原点
	View→Center At Cursor	选择显示中心
	View→Zoom In	放大
	View→Zoom Out	缩小
	View→Zoom To View Entire Sheet	显示全部
	View→Zoom To Area	缩放一个区域
	Edit→Undo	撤销
	Edit→Redo	恢复
	Edit→Cut To Clipboard	剪切
	Edit→Copy To Clipboard	复制
	Edit→Paste To Clipboard	粘贴
	Block Copy	(块) 复制
	Block Move	(块) 移动
	Block Rotate	(块) 旋转
	Block Delete	(块) 删除
	Library→Pick Parts From Libraries	拾取元件或符号


续表


按 钮	对 应 菜 单	功 能
	Library→Make Device	制作元件
	Library→Packaging Tool	封装工具
	Library→Decompose	分解元件
	Tool→Wire Auto Router	自动布线器
	Tools→Search and Tag	查找并标记
	Tools→Property Assignment Tool	属性分配工具
	Design→New Sheet	新建图纸
	Design→ Remove Sheet	移去图纸
	Exit to Parent Sheet	转到主原理图
	Tools→Electrical Rule Check	生成电气规则检查报告

3. 工具箱


-  Selection Mode 按钮：选择模式，可以单击任意元件并编辑元件的属性。
 -  Component Mode 按钮：拾取元件。
 -  Junction Dot Mode 按钮：放置节点可在原理图中标注连接点。
 -  Wire Label Mode 按钮：标注线段或网络名。
 -  Text Script Mode 按钮：输入文本。
 -  Buses Mode 按钮：绘制总线和总线分支。
 -  Subcircuit Mode 按钮：绘制电子块。
 -  Terminals Mode 按钮：在对象选择器中列出各种终端（输入、输出、电源和地等）。
 -  Device Pins Mode 按钮：在对象选择器中列出各种引脚（如普通引脚、时钟引脚、反电压引脚和短接引脚等）。
 -  Graph Mode 按钮：在对象选择器中列出各种仿真分析所需的图表（如模拟图表、数字图表、混合图表和噪声图表等）。
 -  Active Popup Mode 按钮：对设计电路分割仿真时采用此模式。
 -  Generator Mode 按钮：在对象选择器中列出各种激励源（如正弦激励源、脉冲激励源、指数激励源和 FILE 激励源等）。
 -  Probe Mode 按钮：可在原理图中添加探针（如电压探针和电流探针）。
 -  Virtual Instruments Mode 按钮：在对象选择器中列出各种虚拟仪器（如示波器、逻辑分析仪、定时/计数器和模式发生器等）。
- 除上述图标按钮外，系统还提供了 2D 图形模式按钮，可供画线、画弧等。
-  2D Graphics Line Mode 按钮：直线图标，用于创建元件或表示图表时画线。
 -  2D Graphics Box Mode 按钮：方框图标，用于创建元件或表示图表时绘制方框。
 -  2D Graphics Circle Mode 按钮：圆图标，用于创建元件或表示图表时画圆。
 -  2D Graphics Arc Mode 按钮：弧线图标，用于创建元件或表示图表时绘制弧线。
 -  2D Graphics Closed Path Mode 按钮：任意形状图标，用于创建元件或表示图表时绘制任意形状图标。

 2D Graphics Text Mode 按钮: 文本编辑图标, 用于插入各种文字说明。


 2D Graphics Symbols Mode 按钮: 符号图标, 用于选择各种符号器件。


 2D Graphics Markers Mode 按钮: 标记图标, 用于产生各种标记图标。

对于具有方向性的对象, 系统还提供了各种旋转图标按钮 (需要选中对象):

 Rotate Clockwise 按钮: 顺时针方向旋转按钮, 以 90° 偏置改变元件的放置方向。

 Rotate Anti-clockwise 按钮: 逆时针方向旋转按钮, 以 90° 偏置改变元件的放置方向。

 X-mirror 按钮: 水平镜像旋转按钮, 以 Y 轴为对称轴, 按 180° 偏置旋转元件。

 Y-mirror 按钮: 垂直镜像旋转按钮, 以 X 轴为对称轴, 按 180° 偏置旋转元件。



1.4 编辑窗口显示导航

PROTEUS 中提供了两种编辑窗口显示导航工具: 缩放 (Zooming) 工具与改变显示中心 (Center At Cursor) 工具。其中缩放工具用于调整原理图的显示范围, 而改变显示中心工具用于调整图页的显示区域。

1. 缩放

PROTEUS ISIS 中提供了多种放大与缩小原理图的方式。

- ☺ 使用鼠标滚轮缩放原理图 (向前滚动滚轮, 将放大原理图; 向后滚动滚轮, 将缩小原理图)。
- ☺ 使用功能键缩放原理图 (将鼠标指向想要进行缩放的部分, 并按下放大功能键 F6/ 缩小功能键 F7, 编辑窗口将以鼠标指针的位置为中心重新显示)。
- ☺ 按住 Shift 键, 用鼠标左键将期望放大的部分选中, 此时选中的部分将会被放大。(鼠标可在编辑窗口操作, 也可在预览窗口操作。)
- ☺ 使用工具栏 Zoom In (放大)、Zoom Out (缩小)、Zoom All (显示全部) 或 Zoom Area (缩放一个区域) 图标缩放编辑窗口, 如图 1-12 所示。

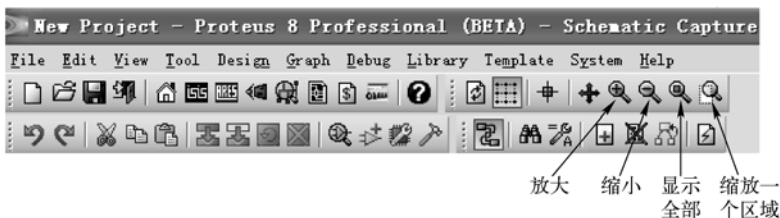


图 1-12 工具栏缩放工具

说明:

F8 键为 Zoom All 的快捷键。按下 F8 键, PROTEUS 将显示整张电路原理图。

2. 改变显示中心

与缩放功能一样, PROTEUS ISIS 中也提供了多种改变编辑窗口显示中心的方式。

- ☺ 在编辑窗口单击滚轮, 然后移动鼠标, 此时编辑窗口的显示中心将随着鼠标的移动

而移动。当出现期望显示的部分时单击鼠标左键，编辑窗口将显示期望的部分。

- ☺ 将鼠标放置在期望显示的部分，按下 F5 键，编辑窗口将显示期望的部分。
- ☺ 在编辑窗口中按下 Shift 键，用鼠标“撞击”边框，可改变编辑窗口的显示中心。
- ☺ 在预览窗口，在期望显示的部分单击鼠标左键，即可改变编辑窗口的显示中心。
- ☺ 使用工具栏 Center At Cursor（改变显示中心）图标改变显示中心，如图 1-13 所示。

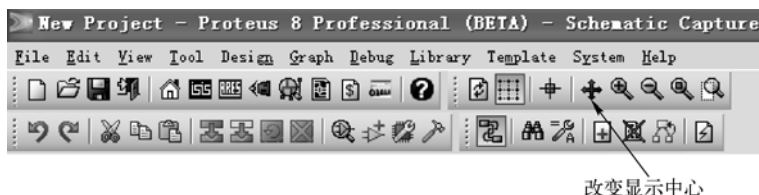


图 1-13 工具栏改变显示中心工具



1.5 编辑窗口的设置

1. 编辑窗口的图纸

在绘制电路时，首先须按照电路的大小选择图纸，PROTEUS ISIS 中提供图纸选项。单击 System→Set Sheet Sizes，将出现如图 1-14 所示的对话框。

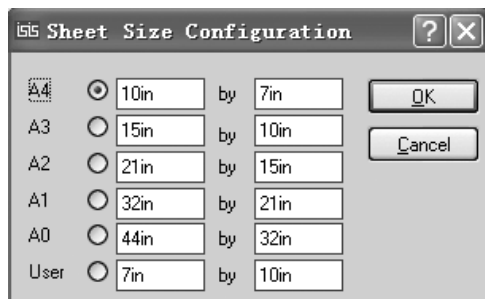


图 1-14 设置图纸大小

对于各种不同应用场合的电路设计，图纸的大小也不一样。系统默认图纸大小为 A4，如用户要将图纸大小更改成为标准 A3 图纸，将 A3 选中，单击“OK”按钮确认即可。

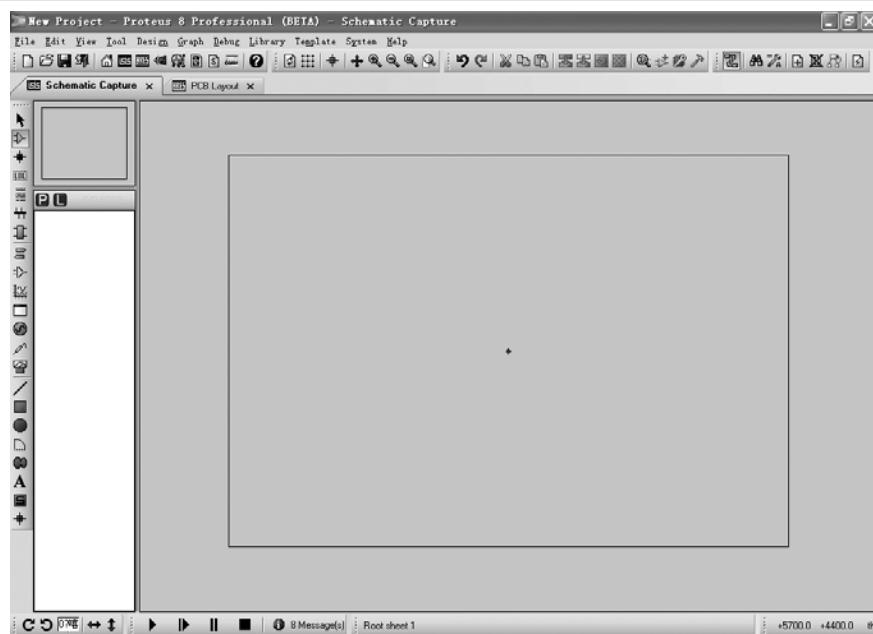
系统所提供的图纸样式有以下几种：

- ☺ 美制：A0、A1、A2、A3、A4，其中 A4 为最小。
- ☺ 用户自定义：User。

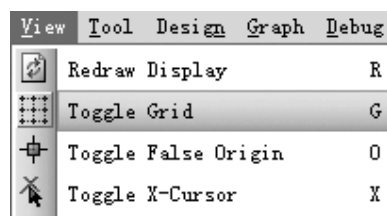
2. 编辑窗口的点状栅格

在设计电路图时，图纸上的点状栅格为放置元件和连接线路带来了很大的帮助，也使电路图中元件的对齐、排列更加方便。

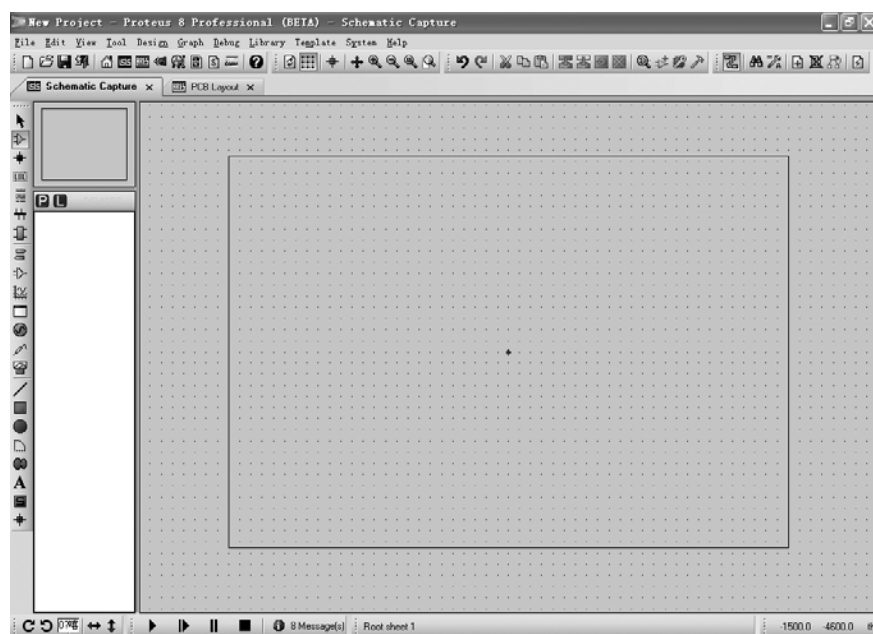
【点状栅格的显示与隐藏】选择 View→Toggle Grid (快捷键：G)设置窗口中的点状栅格的显示与隐藏，如图 1-15 所示。



(a) 隐藏点状栅格



(b) View→Toggle Grid 命令



(c) 显示点状栅格

图 1-15 点状栅格的隐藏与显示

或单击工具栏中的 Grid 图标（如图 1-16 所示），也可实现对点状栅格的操作。

【点状栅格的设置】选择 Template→Set Design Colours 菜单命令，如图 1-17 所示，将弹出如图 1-18 所示的编辑设计默认选项对话框。单击 Grid Colour 选项的下拉式按钮，将弹出调色板，如图 1-19 所示。



点状栅格操作工具

图 1-16 点状栅格操作工具

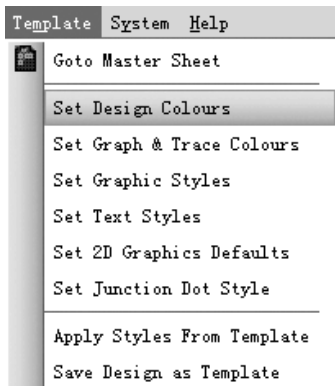


图 1-17 Template→Set Design Colours 菜单命令

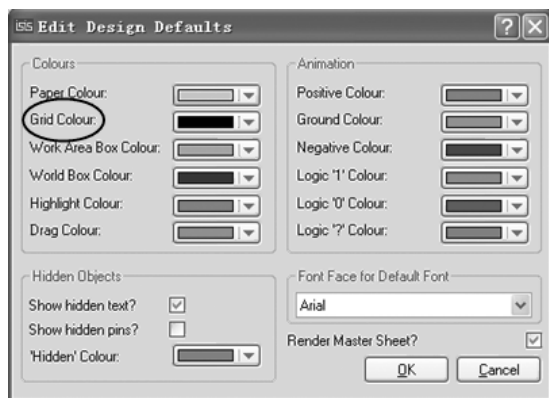


图 1-18 编辑设计默认选项对话框

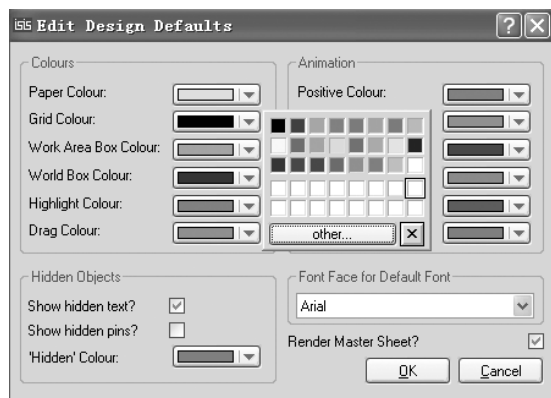


图 1-19 点状栅格格点颜色设置

选择期望的颜色后单击“OK”按钮，即可改变点状栅格格点的颜色。

通过上述对话框可对图形轮廓线（Graph Outline）、底色（Background）、图形标题（Graph Title）、图形文本（Graph Text）等按用户期望的颜色进行设置，同时也可对模拟跟踪曲线（Analogue Traces）、不同类型的数字跟踪曲线（Digital Traces）进行设置。

选择 Template→Set Graphics Styles 菜单命令，编辑图形风格，如图 1-20 所示。

使用这一编辑框可以编辑图形风格，如线型、线宽、线的颜色，以及图形的填充色等。点选 Style 可选择不同的系统图形风格。单击“New”按钮，将弹出如图 1-21 所示的对话框。

用户在 New style's name 文本框中输入新的风格的名称，如 peal，并单击“OK”按钮，将出现如图 1-22 所示的窗口。

在该窗口中用户可自定义图形的风格，如颜色、线型等。选择 Template→Set Text Styles 菜单命令，编辑全局字体风格，如图 1-23 所示。

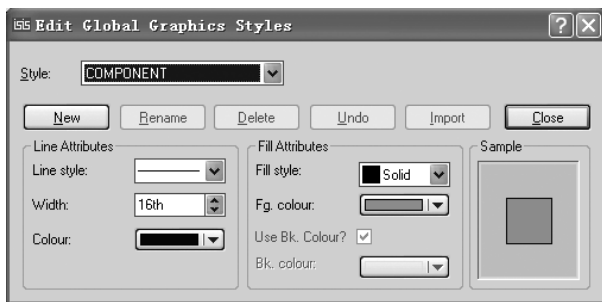


图 1-20 编辑图形风格

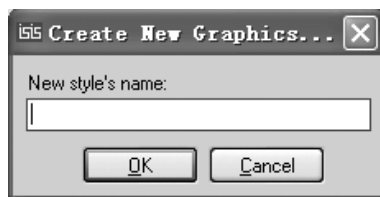


图 1-21 创建新的图形风格对话框

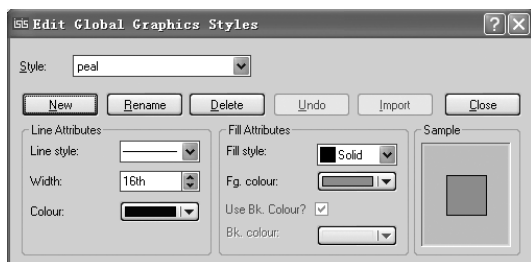


图 1-22 设置新的图形的风格

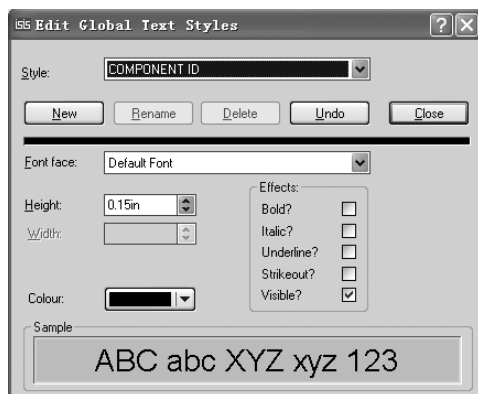


图 1-23 编辑全局字体风格

单击 Font face 的下拉式列表，可从中选择期望的字体，还可设置字体的高度、颜色及是否加粗、倾斜、加下画线等。在 Sample 区域可以预览更改设置后字体的风格。

同理，单击“New”按钮可创建新的图形文本风格。

选择 Template→Set 2D Graphics Defaults 菜单命令，编辑图形字体格式，如图 1-24 所示。出现这一编辑框后，可在 Font face 中选择图形文本的字体类型，在 Text Justification 选择区域可选择字体在文本框中的水平位置、垂直位置，在 Effects 选择区域可选择字体的效果，如加粗、倾斜、加下画线等，而在 Character Sizes 设置区域，可以设置字体的高度和宽度。

选择 Template→Set Junction Dot Style 菜单命令，编辑交点，如图 1-25 所示。

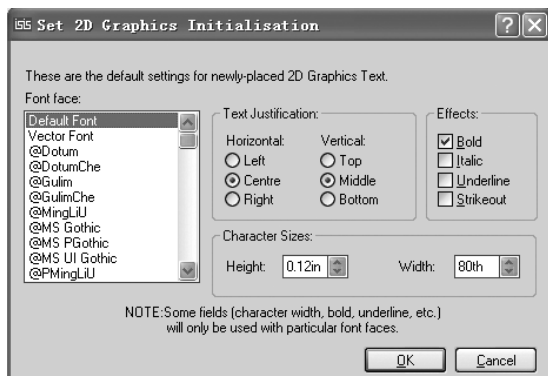


图 1-24 编辑图形字体格式

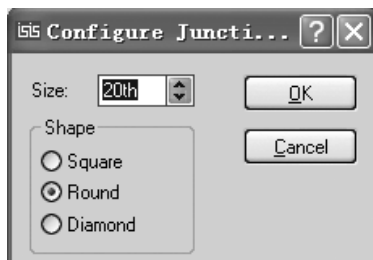


图 1-25 编辑交点

可以设置交点的大小及其形状。单击“OK”按钮，即可完成对交点的设置。

注意：

模板的改变仅仅只影响当前运行的 ISIS，尽管这些模板有可能被保存并且在别的设计中调用。为了使下次开始一个设计的时候这个改变依然有效，用户必须用保存为默认模板命令去更新默认的模板，这个命令在模板菜单下，为 Template→Save Design as Template。

3. 文本编辑器的设置

选择 System→Set Text Editor 菜单命令，将出现如图 1-26 所示的对话框。



图 1-26 设置文本格式

可以对文本的字体、字形、大小、效果、颜色等进行设置。

4. 设置系统运行环境

选择 System→Set Environment 命令，即可打开系统环境设置对话框，如图 1-27 所示。

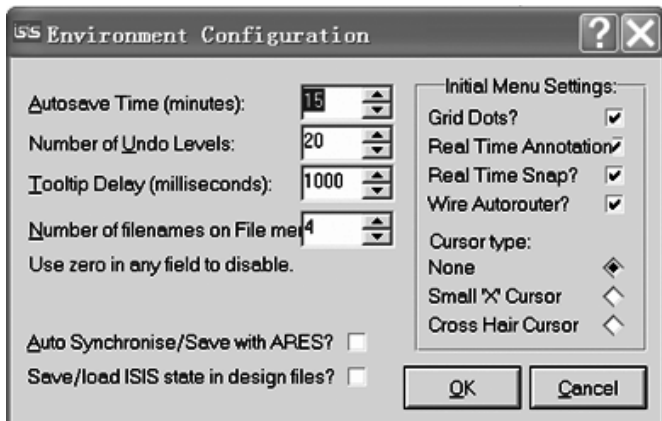


图 1-27 系统环境设置对话框

选项区域主要包括如下设置:

- ☺ Autosave Time (minutes): 系统自动保存时间设置 (min)。
- ☺ Number of Undo Levels: 可撤销操作的数量设置。
- ☺ Tooltip Delay (milliseconds): 工具提示延时 (ms)。
- ☺ Auto Synchronise/Save with ARES: 自动同步/保存 ARES。
- ☺ Save/load ISIS state in design files: 在设计文档中加载/保存 ISIS 状态。
- ☺ Initial Menu Settings: 初始菜单设置。
 - ▮ Grid Dots: 是否显示格点;
 - ▮ Real Time Annotation: 是否进行实时标注;
 - ▮ Real Time Snap: 是否进行实时捕捉;
 - ▮ Wire Autorouter: 是否开启线路自动路径器。
- ☺ Cursor type: 指针类型。
 - ▮ None: 无;
 - ▮ Small “X” Cursor: 小 “X” 型指针;
 - ▮ Cross Hair Cursor: 交叉十字指针。

【实时捕捉 (Real Time Snap)】即当鼠标指针指向引脚末端或者导线时, 鼠标指针将会捕捉到这些对象, 这种功能称为实时捕捉。该功能可以使用户方便地实现导线与引脚的连接。可以通过Tools菜单的Real Time Snap 命令或者是“Ctrl+S”快捷键切换该功能。

【线路自动路径器 (WAR)】线路自动路径器 (WAR) 为用户省去了必须标明每根线具体路径的麻烦。该功能默认是打开的。

如果用户只是在两个连接点左击, WAR将选择一个合适的线径。如果用户单击了一个连接点, 然后单击一个或几个非连接点的位置, ISIS将认为用户在手工指定线的路径, 因此会将用户单击处作为路径的拐点。路径是通过左击另一个连接点来完成的。

WAR可通过使用工具菜单里的WAR命令来关闭。这一功能在用户给出期望连接的两个点间的对角线时是很有用的。

5. 设置键盘快捷方式

选择 System→Set Keyboard Mapping 命令, 即可打开键盘快捷方式设置对话框, 如图 1-28 所示。

使用这一对话框可修改系统所定义的菜单命令的快捷方式。

其中, 单击 Command Groups 栏中的箭头, 可选择相应的菜单, 同时在列表栏中显示菜单下的可用的命令 (Available Commands)。在列表栏下方的说明栏中显示所选中的命令的意义。而 Key sequence for selected command 栏中显示所选命令的键盘快捷方式。使用 “Assign” 和 “Unassign” 按钮可编辑或删除系统设置的快捷方式。

同时选中 Options 选项卡, 将出现如图 1-29 所示的菜单。

使用其中的 Reset to default map 选项, 即可恢复系统的默认设置。而 Export to file 可将上述键盘快捷方式导出到文件, Import from file 为从文件导入。

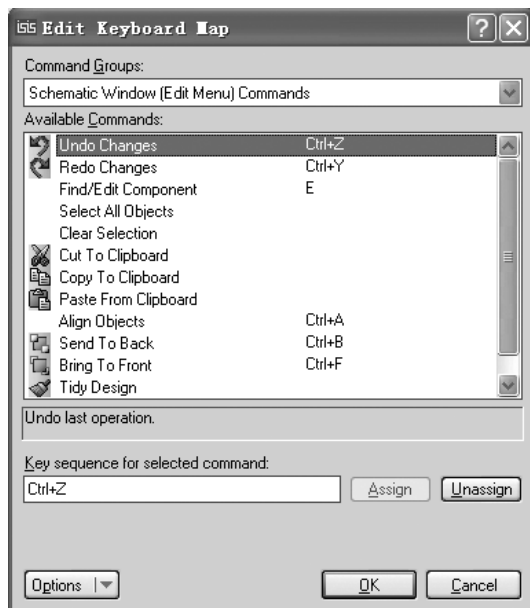


图 1-28 键盘快捷方式设置对话框

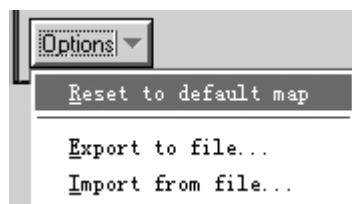


图 1-29 Options 选项卡菜单

第2章 PROTEUS ISIS 原理图设计



2.1 PROTEUS ISIS 原理图输入流程

电路设计的第一步为原理图输入。PROTEUS ISIS 原理图输入流程如图 2-1 所示。

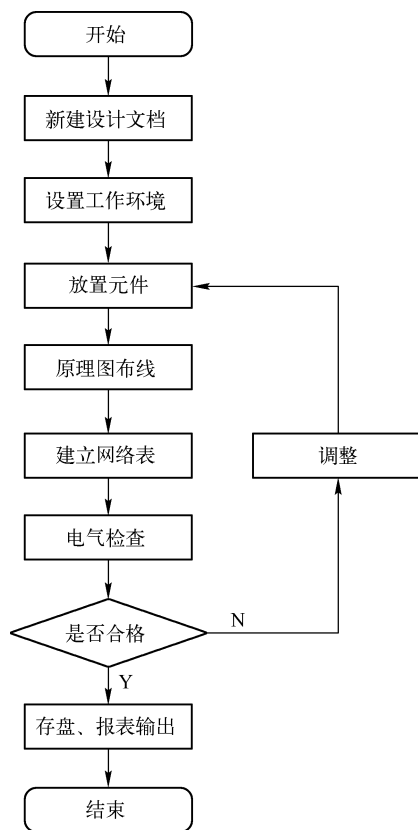


图 2-1 原理图输入流程图

原理图的具体设计步骤如下：

1) 新建设计文档 在进入原理图设计之前，首先要构思好原理图，即必须知道所设计的项目需要哪些电路来完成，用何种模板，然后用 PROTEUS ISIS 编辑环境来画出电路原理图。

2) 设置工作环境 根据实际电路的复杂程度来设置图纸的大小、注释的风格等。在电路图设计的整个过程中，对图纸的大小都可以不断地调整，设置合适的图纸大小是完成原理图设计的第一步。

3) 放置元件 根据需要从元件库中添加相应的类，然后从添加元件对话框中选取需要添加的元件，将其布置到图纸的合适位置，并对元件的名称、标注进行设定，根据元件之间的走线等联系对元件在工作平面上的位置进行调整和修改，使得原理图美观而且易懂。

4) 原理图布线 根据实际电路的需要，利用 PROTEUS ISIS 编辑环境所提供的各种工具、指令进行布线，将工作平面上的器件用导线连接起来，构成一幅完整的电路原理图。

5) 建立网络表 在完成上述的步骤之后，即可以看到一张完整的电路图了，但是要完成电路板的设计，就需要生成一个网络表文件。网络表是电路板与

电路原理图之间的纽带。

6) 电气检查 当完成原理图布线后，利用 PROTEUS ISIS 编辑环境所提供的电气规则检查命令对设计进行检查，并根据系统提供的错误检查报告修改原理图。

7) 调整 如果原理图已通过电气规则检测，那么原理图的设计就完成了；但是对于一般电路设计而言，尤其是较大的项目，通常需要对电路进行多次修改才能够通过电气规则检测。

8) 存盘、报表输出 PROTEUS ISIS 提供了多种报表输出格式，同时可以对设计好的

原理图和报表进行存盘及输出打印。



2.2 原理图设计方法与步骤

为了更直观地说明电路原理图的设计方法和步骤，下面就以图 2-2 所示的简单电路为例，介绍 PROTEUS ISIS 电路原理图的设计方法和步骤。

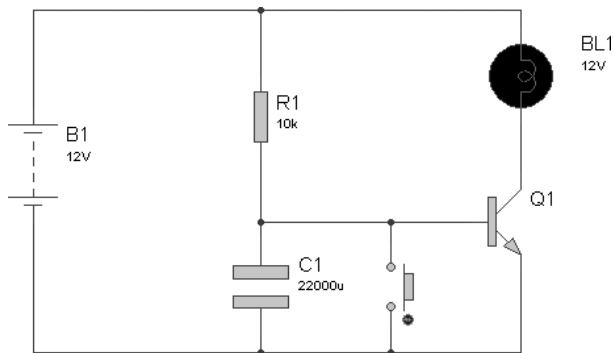


图 2-2 PROTEUS ISIS 原理图输入示例电路

1. 创建新设计文档

首先进入 PROTEUS ISIS 编辑环境。单击主菜单 File→New Design，在弹出的模板对话框中选择 Default 模板，并将新建的设计保存在 F 盘根目录下，保存文件名为：New Project。新的设计文档如图 2-3 所示。

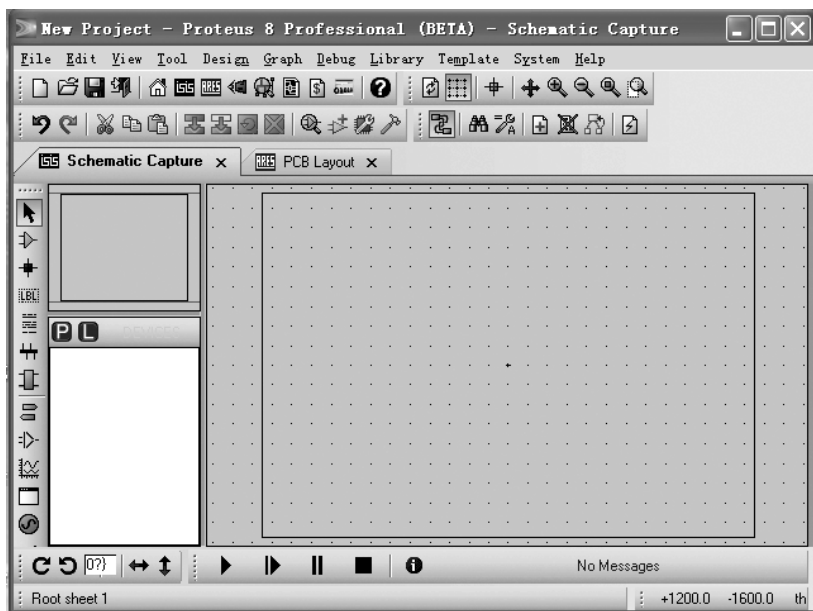


图 2-3 New Project 设计文档

2. 设置工作环境

选择 **Template** 菜单, 对工作环境进行设置。在本例中, 仅对图纸进行设置, 其他项目使用系统默认的设置。选择 **System**→**Set Sheet Sizes** 菜单命令, 在出现的对话框中, 将 **A4** 选中, 单击“OK”按钮确认, 即可完成页面的设置, 如图 2-4 所示。

PROTEUS ISIS 具有友好的用户界面及强大的原理图编辑功能。在 ISIS 原理图编辑窗口, 系统提供了两种可视工具:

☺ 当鼠标掠过元件、符号、图形等对象时, 将出现围绕对象的虚线框, 如图 2-5 所示。

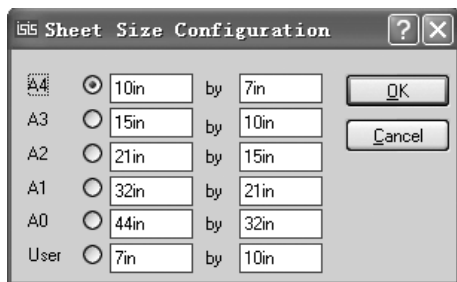


图 2-4 图纸设置

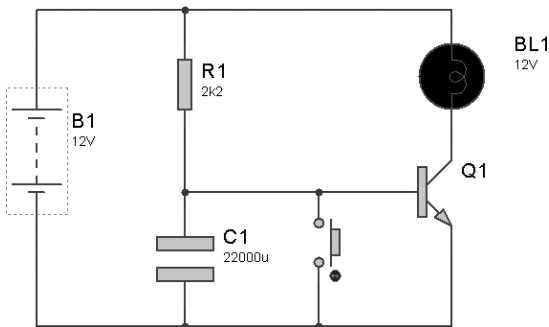


图 2-5 鼠标掠过元件时出现围绕对象的虚线框

☺ 鼠标对界面具有智能识别功能, 即鼠标会自动根据功能改变显示式样。

鼠标掠过元件出现虚线框提示用户鼠标可对当前元件进行操作, 而不同的鼠标样式提示用户当前位置单击鼠标将会出现的操作。

不同的鼠标样式代表用户可进行不同的操作:

☞ 标准指针: 用于选择操作模式。

☞ 放置指针: 单击鼠标左键放置对象。

☞ “热”画线指针: 单击鼠标左键, 开始画线或结束画线。“热”指在鼠标操作电路图时, 在电路中可放置“线”的起始点或结束点的位置, 将显示上述图标。

☞ “热”画总线指针: 单击鼠标左键, 开始画总线或结束画总线。“热”指在鼠标操作电路图时, 在电路中可放置“总线”的起始点或结束点的位置, 将显示上述图标。

☞ 当对象上出现此图标后, 单击鼠标左键, 对象被选中。

☞ 当对象上出现此图标后, 按下鼠标左键并拖动鼠标, 对象可被拖到期望的位置。

☞ 此鼠标样式出现在线段上。当线段上出现此图标后, 按下鼠标左键并拖动鼠标, 线段可被拖到期望的位置。

☞ 当出现此图标时, 单击鼠标左键可为对象分配属性。(使用 **Tool**→**Property Assignment Tool** 时将出现此鼠标样式。)

其他的鼠标样式及其功能将在电路绘制中介绍。

3. 查找元件

绘制原理图的首要任务是从元件库选取绘制电路所需元件。PROTEUS ISIS 提供两种从元件库选取元件的方法:

1) **直接查找** 点选对象选择器顶端左侧“P”按钮, 如图 2-6 所示; 或使用库浏览图

标的键盘快捷方式：“P”（在英文输入法下）；在原理图编辑窗口单击鼠标右键，将弹出右键菜单，选择 Place→Component→From Libraries 命令，如图 2-7 所示。



图 2-6 从对象选择器选取库元件

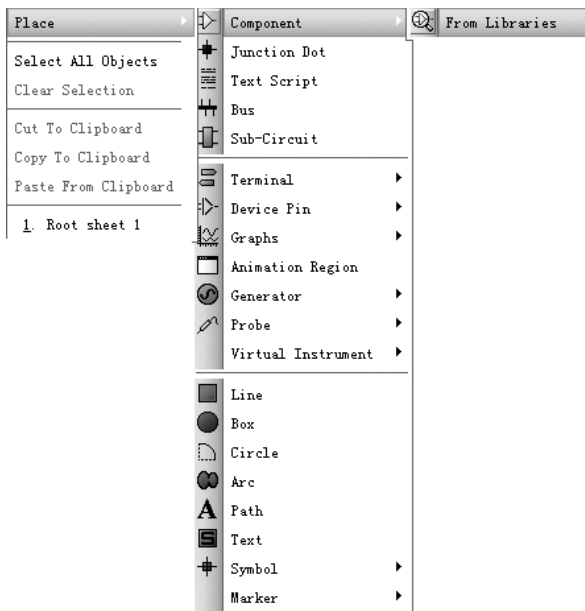


图 2-7 从右键菜单选取库元件

执行上述任一操作，都将弹出如图 2-8 所示的器件库浏览对话框。

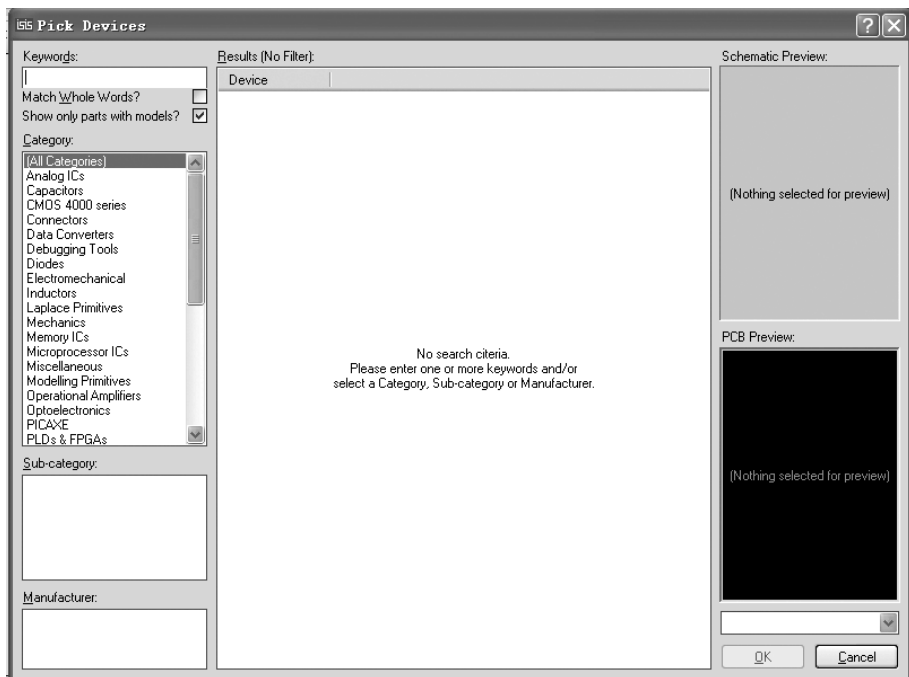


图 2-8 器件库浏览对话框

在元件库查找期望的元件。PROTEUS ISIS 提供了多种查找元件的方法。当原理图给出

元件名时, 在 Keywords 区域输入元件名, 如“BATTERY”, 则在 Results 区域显示出元件库中元件名或元件描述中带有“BATTERY”的元件, 如图 2-9 所示。

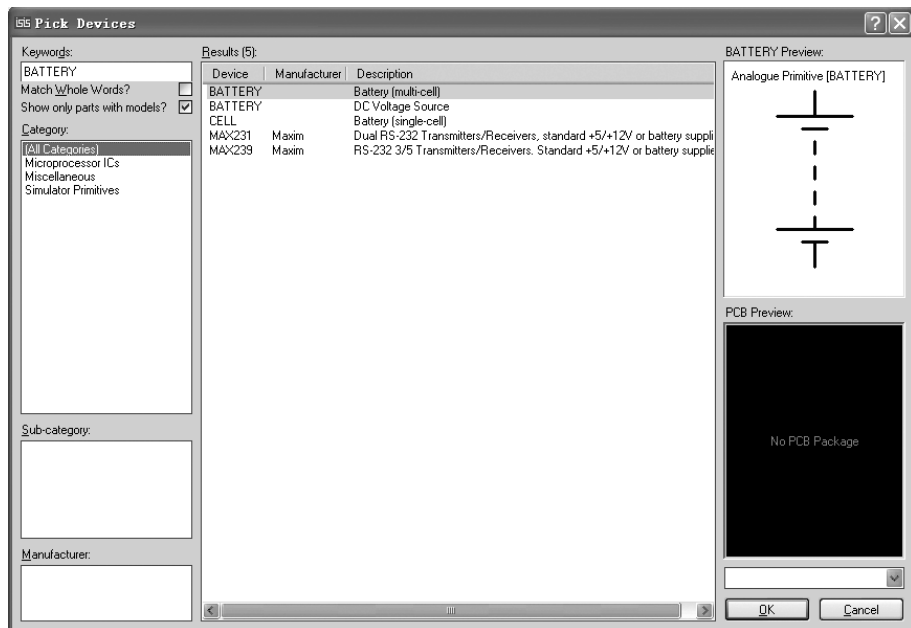


图 2-9 在 Keywords 区域输入元件“BATTERY”后系统查找的结果

此时用户可根据元件所属类别、子类及生产厂家进一步查找所需元件, 如图 2-10 所示。

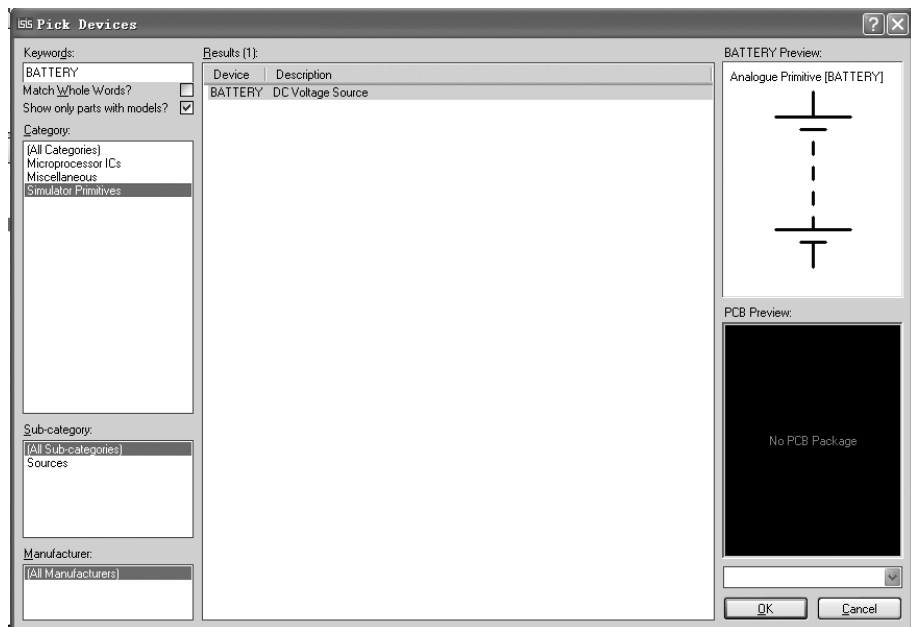


图 2-10 根据元件所属类别进一步查找所需元件

在 Results 列表区单击鼠标右键, 将弹出右键菜单, 如图 2-11 所示。

选择相应的选项将在 Results 列表区标题栏增加相应的信息, 如点选 Manufacturer 选

项，如图 2-12 所示。

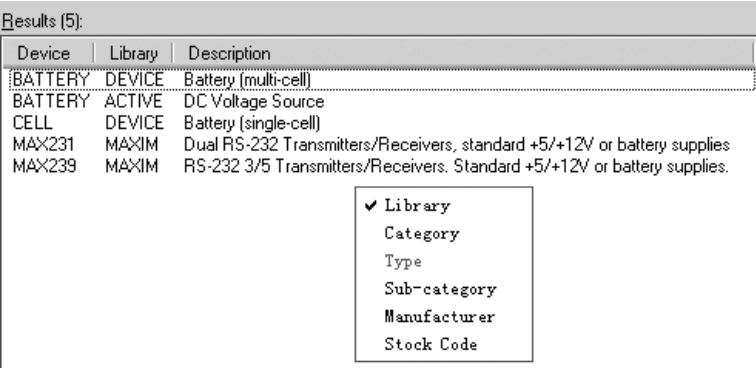


图 2-11 Results 列表区鼠标右键菜单

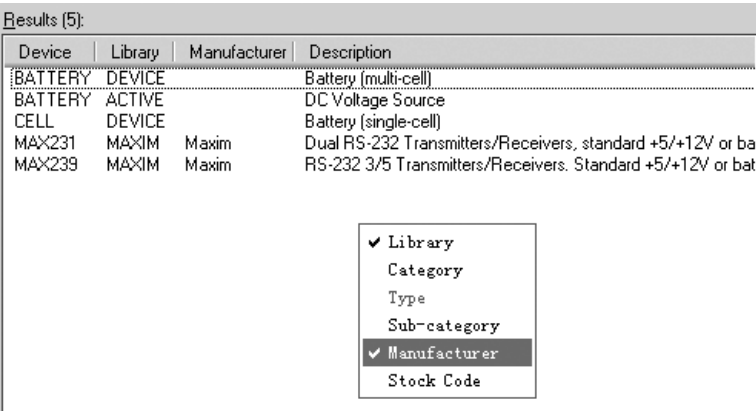


图 2-12 增加元件信息

此时在 Results 列表区将增加 Manufacturer（生产厂家）的信息，如图 2-13 所示。在结果列表的所需元件“BATTERY”上双击鼠标左键，元件将出现在对象选择器中，如图 2-14 所示。

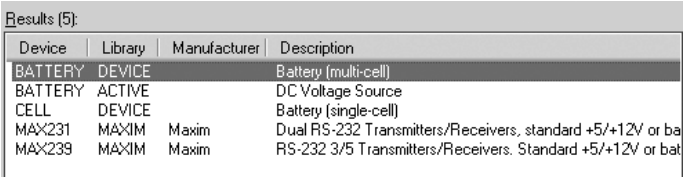


图 2-13 增加信息后的显示结果

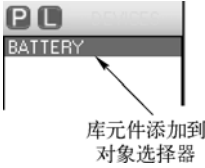


图 2-14 将元件添加到对象选择器

选取元件的其他方法：以选取 LAMP（荧光灯）、CAPACITOR（电容）、RESISTOR（电阻）、NPN（晶体管）和 BUTTON（触电开关）为例。

【通过相关关键字选取元件】在 Keywords 区域输入“LAMP”，此时 Results 列表区将出现如图 2-15 所示信息。

选取其中的 LAMP 即可满足电路设计要求。

【按照元件的逻辑命名习惯查找元件】在 Keywords 区域输入“CAPACITOR”，此时

Results 列表区将出现如图 2-16 所示信息。

Results (16):			
Device	Library	Manufacturer	Description
4511	CMOS		BCD To 7-Segment Latch/Decoder/Driver
4511.IEC	CMOS		BCD To 7-Segment Latch/Decoder/Driver
AD8036	ANALOGD	Analog Devices	Single, Unity Gain Stable - Low Distortion, V
AD8036AN	ANALOGD	Analog Devices	Low Distortion, Wide Bandwidth Voltage Fe
AD8036AR	ANALOGD	Analog Devices	Low Distortion, Wide Bandwidth Voltage Fe
AD8037AN	ANALOGD	Analog Devices	Low Distortion, Wide Bandwidth Voltage Fe
AD8037AR	ANALOGD	Analog Devices	Low Distortion, Wide Bandwidth Voltage Fe
IRGB14C40L	IRIGBT	International Rectifier	430V, 20A @ 25°C, 125W, Internal Clamp,
IRGS14C40L	IRIGBT	International Rectifier	430V, 20A @ 25°C, 125W, Internal Clamp,
IRGSL14C40L	IRIGBT	International Rectifier	430V, 20A @ 25°C, 125W, Internal Clamp,
LAMP	ACTIVE		Animated Light Bulb
STGB10NB37LZ	STIGBT	SGS-Thompson	N-CHANNEL CLAMPED 10A - D2PAK Inte
STGB10NB37LZT4	STIGBT	SGS-Thompson	N-CHANNEL CLAMPED 10A - D2PAK Inte
STGB20NB32LZ	STIGBT	SGS-Thompson	N-CHANNEL CLAMPED 20A - D2PAK Inte
STGB20NB32LZ-1	STIGBT	SGS-Thompson	N-CHANNEL CLAMPED 20A - D2PAK Inter
STGB7NB40LZ	STIGBT	SGS-Thompson	N-CHANNEL CLAMPED 14A D2PAK Inter

图 2-15 在 Keywords 区域输入“LAMP”时，Results 列表区出现的信息

Results (2):		
Device	Library	Description
CAPACITOR	ASIMMDLS	Capacitor primitive
CAPACITOR	ACTIVE	Animated Capacitor model

图 2-16 在 Keywords 区域输入“CAPACITOR”时，Results 列表区出现的信息

同样，用户可从出现的列表选取 RESISTOR（电阻）、NPN（晶体管）和 BUTTON（触电开关）以满足电路设计要求。

【通过索引系统查找库元件】当用户不确定元件的名称或不清楚元件的描述时，可采用这一方法。首先清除 Keywords 区域的内容，然后选择 Category 目录中的 Resistors 类，如图 2-17 所示。

2) 复合查找方式 首先在 Keywords 区域输入“2K2”，然后选择 Category 目录中的 Resistors 类，如图 2-18 所示。



图 2-17 清除 Keywords 区域的内容

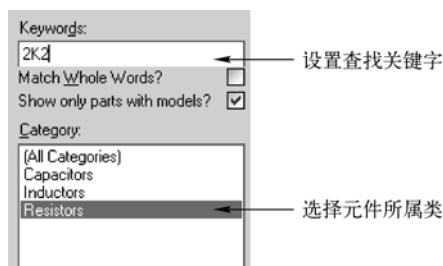


图 2-18 在 Keywords 区域键入“2K2”

此时将在 Results 列表区出现如图 2-19 所示信息。

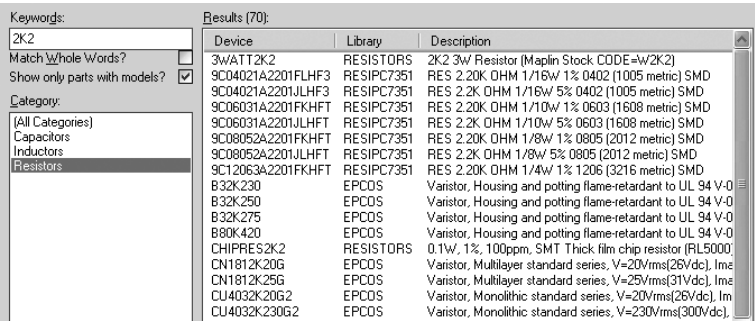


图 2-19 Results 列表区列出所属 Resistors 类中包含 2K2 关键字的元件

采用上述方法，用户可以快速查找到设计电路所需元件。
其他元件名称、所属类、子类如表 2-1 所示。

表 2-1 例图原理图的元件列表

元 件 名 称	所 属 类	所 属 子 类
LAMP	Optoelectronics	Lamps
CAPACITOR	Capacitors	Animated
BATTERY	Simulator Primitives	Sources
NPN	Modeling Primitives	Analog（SPICE）
BUTTON	Switches & Relays	Switches
RESISTOR	Modeling Primitives	Analog（SPICE）

4. PROTEUS ISIS 编辑窗口放置元件

将添加到对象选择器中的元件放置到原理图编辑窗口。

1) 放置元件 在 PROTEUS ISIS 中放置元件的步骤如下：

设置 PROTEUS ISIS 为元件模式，即元件图标被选择，如图 2-20 所示。在对象选择器中，用鼠标左键单击 BATTERY 元件，即选中 BATTERY 元件，此时在预览窗口出现所选器件的外观，同时状态栏显示对象选择器及预览窗口的状态，如图 2-21 所示。

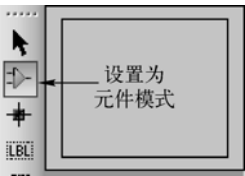


图 2-20 PROTEUS ISIS 设置为元件模式

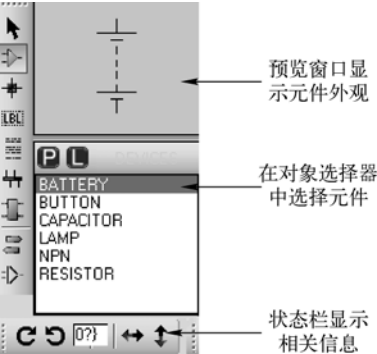


图 2-21 在对象选择器中选择 BATTERY 元件

预览窗口在显示出元件外观的同时，也显示出元件的方向。用户可使用旋转按钮或翻转按钮改变元件的方向，如图 2-22 所示。图中预览窗口显示的元件为执行顺时针旋转后的方向。将鼠标指针移向编辑窗口，并单击鼠标左键，此时电源轮廓出现在鼠标下方，如图 2-23 所示。

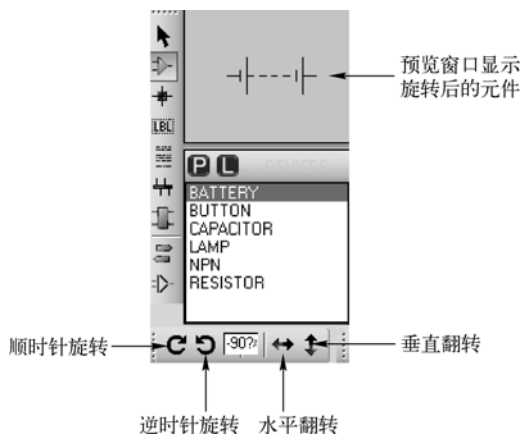


图 2-22 预览窗口显示翻转后的元件

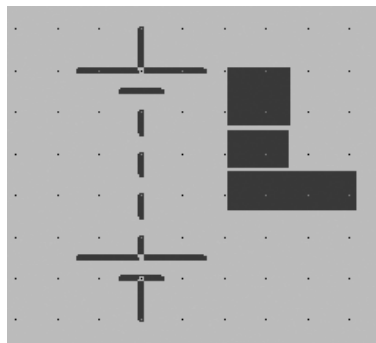


图 2-23 编辑窗口出现电源的轮廓 (BATTERY)

这一轮廓将随着鼠标在编辑窗口移动而移动。在期望放置鼠标的位置单击鼠标左键，元件将放置到编辑窗口，如图 2-24 所示。

2) 其他元件的放置 图 2-2 中，R1 电阻为垂直放置，因此首先放置一垂直放置的电阻。在对象选择器中选择 RESISTOR 电阻，则在编辑窗口出现如图 2-25 所示的电阻外观。

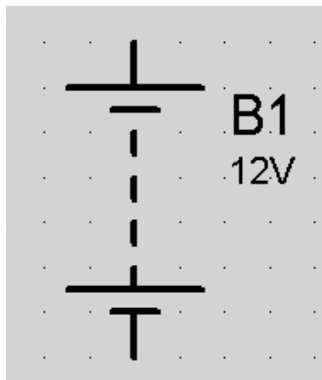


图 2-24 元件 BATTERY 放置到编辑窗口

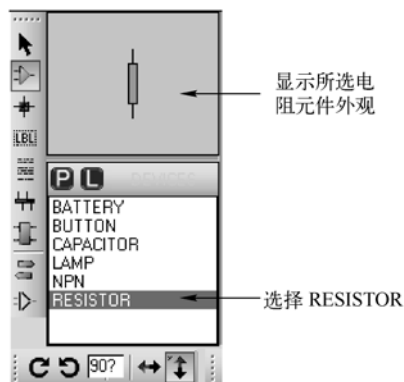


图 2-25 选择 MINRES1K 电阻

在编辑窗口期望放置电阻 R1 的位置双击鼠标左键，R1 电阻将放置到编辑窗口，如图 2-26 所示。

在移动标记对象时，可使用小键盘区的“+”、“-”按钮旋转对象。按照上述方式放置其他元件，如图 2-27 所示。

5. 编辑元件

在放置元件以后，通过选中相应的元件，再单击鼠标左键，即可打开元件的编辑对话框。下面以 LAMP 元件的编辑对话框为例，详细介绍元件的编辑方式。

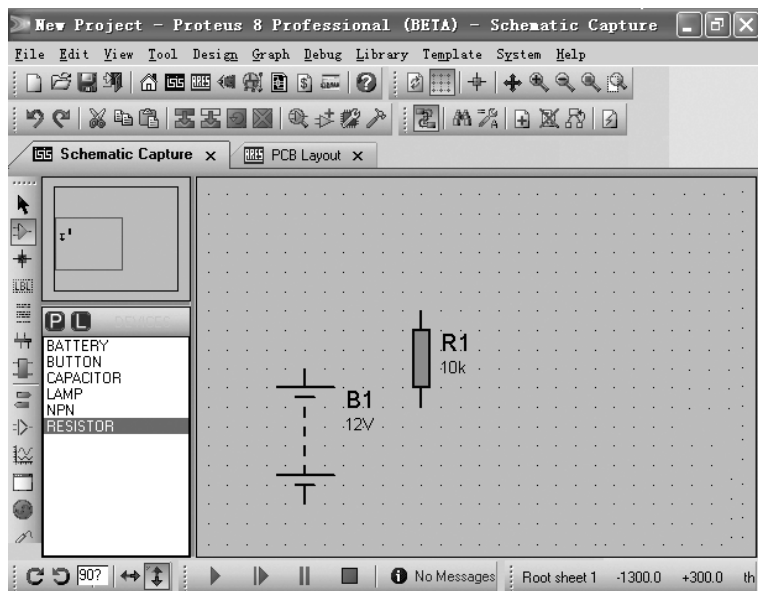


图 2-26 在编辑窗口放置电阻 R1

LAMP 元件编辑步骤如下:

- (1) 用鼠标指向 LAMP 元件, 并单击右键。此时, LAMP 元件将高亮显示。
- (2) 用鼠标左键单击 LAMP 元件, 将弹出如图 2-28 所示 LAMP 编辑元件对话框。

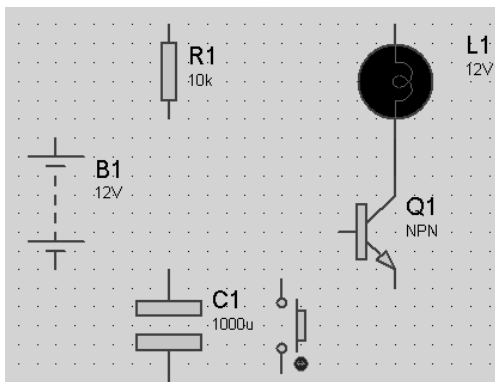


图 2-27 元件放置后

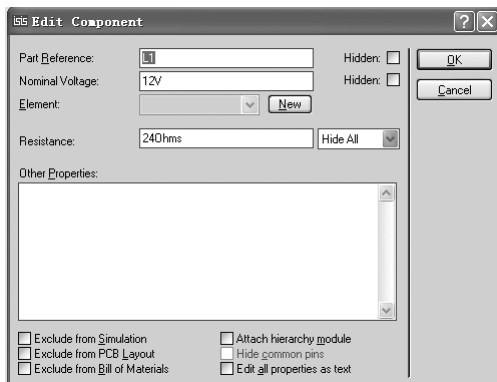


图 2-28 LAMP 编辑元件对话框

在此对话框中包含如下项目:

- ☺ Part Reference: 列出了元件在原理图中的参考号;
- ☺ Hidden: 复选框, 选择元件参考号是否出现在原理图中;
- ☺ Nominal Voltage: LAMP 电压标称值;
- ☺ Resistance: LAMP 阻抗。

在这一元件的编辑中, 设置 LAMP 的电压标称值为 12V。

- (3) 单击“OK”按钮, 元件的编辑结束。

按照上述步骤, 分别编辑 BATTERY 的电压值为 12V; C1 的 Capacitance 为 22 000 μ , R1 的 Resistance (Ohms) 为 2k2。编辑后的原理图如图 2-29 所示。

6. 原理图连线

PROTEUS ISIS 编辑环境没有提供画线工具。这是因为 ISIS 具有智能化，在想要画线的时候进行自动检测。

在两个元件间进行连线的步骤如下：

(1) 左击第一个对象连接点。

(2) 如果你想让 ISIS 自动定出走线路径，只需左击另一个连接点。另一方面，如果你想自己决定走线路径，只需在拐点处单击鼠标左键即可。

(3) 在此过程的任何一个阶段，你都可以按 Esc 键来放弃画线。

按照上述步骤，分别将 LAMP、BATTERY 等连接。连接后的原理图如图 2-30 所示。

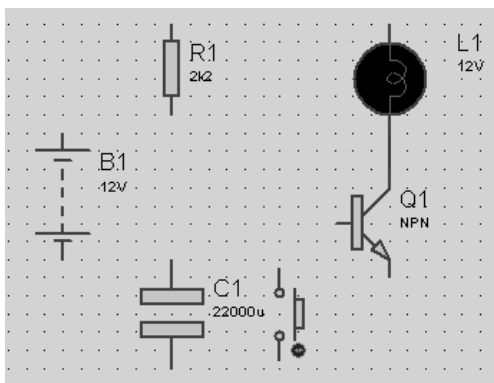


图 2-29 编辑后的原理图

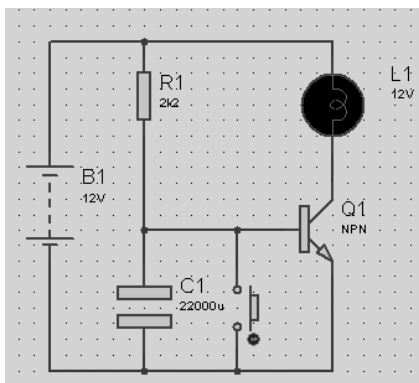


图 2-30 连接后的原理图

7. 建立网络表

彼此连接在一起的一组元件引脚称为网络 (net)。例如例图中 BATTERY 的正极与 R1 的上边的引脚连在一起，称为网络。

选择 Tool→Netlist Compiler 菜单命令，将出现网络表配置对话框，如图 2-31 所示。在这一对话框中，可设置网络表的输出形式、模式、范围、深度及格式。这里按其默认设置输出网络表。输出的网络表如图 2-32 所示。



图 2-31 网络表配置对话框

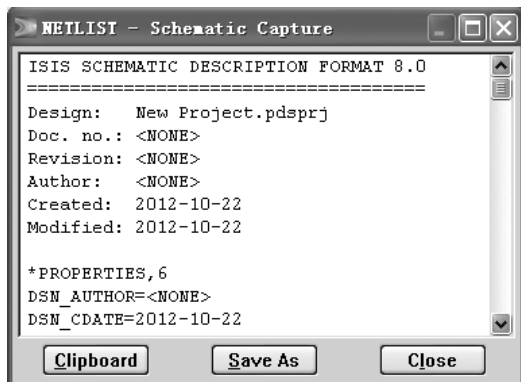


图 2-32 输出的网络表

8. 原理图电气规则检测

选择 Tool→Electrical Rule Check 菜单命令, 将出现检测报告单, 如图 2-33 所示。

在这一规则检测报告单中, 系统提示用户网络表已生成, 并且无电气错误。即用户可执行下一步操作。

9. 存盘及报表输出

将设计好的原理图文件, 使用 File→Save Design 菜单命令存盘。同时, 可使用 Tool→Bill of Materials 菜单命令输出 BOM 文档。至此, 一个简单的原理图设计完成。

按下列步骤操作, 进一步熟悉在 PROTEUS ISIS 编辑窗口放置元件:

(1) 在电源 BATTERY 上单击鼠标左键后, 按下鼠标左键, 并移动鼠标, 此时元件将随着鼠标的移动而移动, 在期望元件放置的位置释放鼠标。

(2) 在电源 BATTERY 上单击鼠标右键, 在弹出的右键菜单中选择 Rotate Clockwise 命令, 旋转元件。

(3) 使用小键盘区的“+”、“-”按钮将元件恢复到原位。

(4) 在编辑窗口的空白处单击鼠标左键, 取消对运放的选择。

(5) 在 MINRES1K 元件上单击鼠标右键, 在弹出的右键菜单中选择 Drag Object 命令, 移动鼠标, 此时电阻轮廓将随着鼠标的移动而移动, 在期望元件放置的位置单击鼠标左键放置对象。

(6) 在编辑窗口单击鼠标右键, 在弹出的右键菜单中选择 Clear Selection 命令, 取消对对象的选择。

(7) 从编辑窗口的左上部开始, 按下鼠标左键, 并拖动鼠标, 此时在编辑窗口将出现一个方框, 释放鼠标, 则方框中的对象被选中。

(8) 使用方框周围的手柄, 将方框调整到合适的尺寸, 即方框内只包含期望选中的对象。

(9) 将鼠标放置在方框中按下鼠标左键, 移动鼠标, 则方框及其选中的对象将随着鼠标的移动一起移动, 在期望放置的位置释放鼠标, 此时方框及其选中的对象将出现在新的位置。

(10) 在空白处单击鼠标左键, 清除对对象的选择。

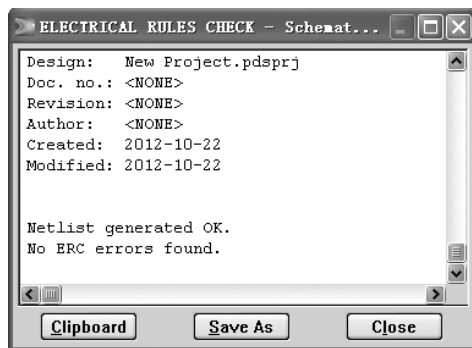


图 2-33 电气规则检测报告单



2.3 PROTEUS ISIS 编辑窗口连接端子

在多数原理图中需要添加电源、地、输入、输出、总线等连接端子。下面介绍一下如何添加、编辑、放置连接端子。

1. 添加连接端子

选择“Terminal”图标，此时在对象选择器中列出可用的端子类型，如图 2-34 所示。

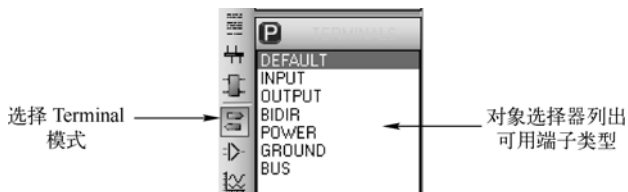


图 2-34 选择“Terminal”图标时，对象选择器中列出可用端子类型

将端子放置到编辑窗口，并将其与运算放大器 741 连接。

选择对象选择器中的“POWER”端子，查看预览窗口，调整端子的方向。调整后的方向如图 2-35 所示，并将其放置到 741 的引脚 7 之上。

2. 编辑连接端子

PROTEUS ISIS 中提供了多种编辑元件的方法，用户可采用下述方法编辑端子：

- ☺ 双击端子；
- ☺ 在端子上单击鼠标右键，在弹出的右键菜单中选择 Edit Properties 选项；
- ☺ 设置编辑窗口为 Selection 模式，在端子上单击鼠标左键，端子将以高亮形式显示，然后单击鼠标右键，弹出右键菜单，选择右键菜单中的 Edit Properties 选项。

其中使用右键菜单编辑端子属性的方法如图 2-36 所示。

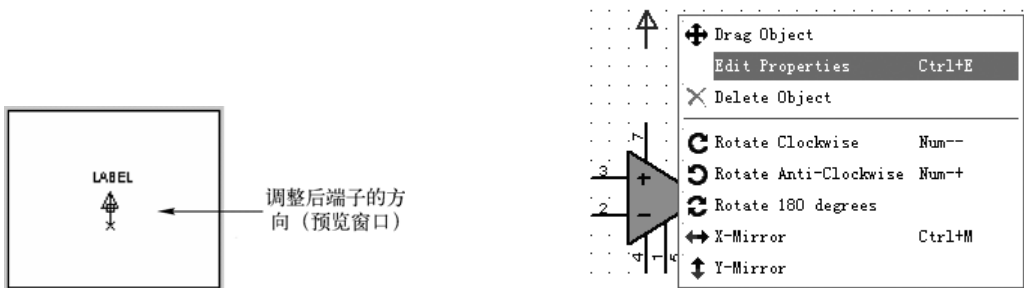


图 2-35 调整后端子的方向

图 2-36 采用右键菜单编辑端子属性

此时系统将弹出编辑端子属性对话框，在 String 文本框中输入+12V，如图 2-37 所示。单击“OK”按钮，退出编辑窗口，完成对端子的编辑，如图 2-38 所示。

3. 放置连接端子

将端子+12V 与运算放大器 741 的 7 引脚相连。

(1) 参照上述方式放置、编辑-12V 端子，并将其与运算放大器 741 的 4 引脚相连。

(2) 选择对象选择器中的“GROUND”端子，查看预览窗口，调整端子的方向，将其放置到电阻 R2 下方，如图 2-39 所示，并将其与 R2 的下端引脚相连。

(3) 选择对象选择器中的“DEFAULT”端子，在编辑窗口放置两个默认端子，分别编辑其端子名为“DAC1”与“DAC2”，如图 2-40 所示。

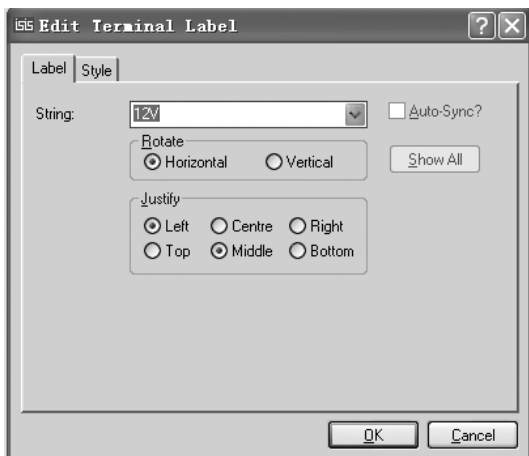


图 2-37 编辑端子属性对话框

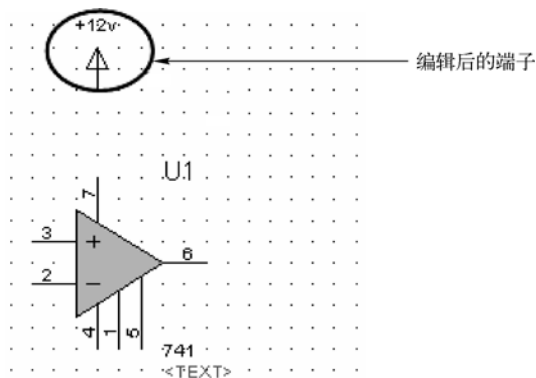


图 2-38 编辑后的端子

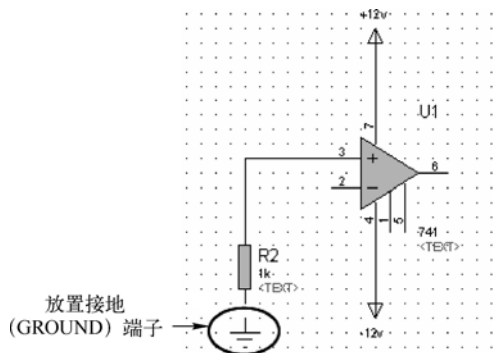


图 2-39 放置“GROUND”端子



图 2-40 放置并编辑默认端子

(4) 按照图 2-41 所示连接电路。

在连线交汇点，PROTEUS ISIS 将自动放置连接点。

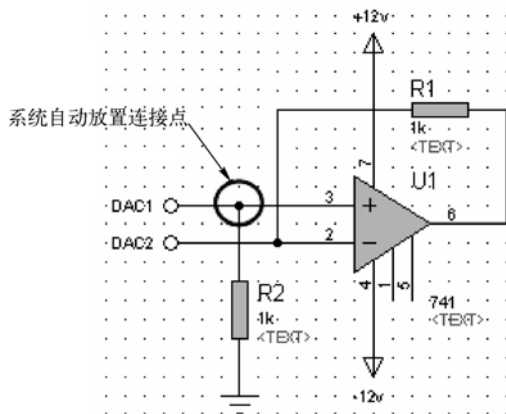


图 2-41 连接后的电路

第3章 电路分析

PROTEUS VSM 中的整个电路分析是在 ISIS 原理图设计模块下延续下来的，原理图中，电路激励、虚拟仪器、曲线图以及直接布置在线路上的探针一起出现在电路中。任何时候都能通过按下运行按钮或空格键对电路进行仿真。

PROTEUS VSM 有两种截然不同的仿真方式：交互式仿真和基于图表的仿真。交互式仿真检验用户所设计的电路是否能正常工作；基于图表的仿真用来研究电路的工作状态和进行细节的测量。

PROTEUS VSM 可以仿真模拟电路、数字电路、模数混合电路，以及微处理器的协同仿真。在下面将依次详细介绍 PROTEUS VSM 中的电路分析。



3.1 PROTEUS ISIS 激励源

激励源为电路提供输入信号，并允许使用者对它的参量进行设置。PROTEUS ISIS 为用户提供了如表 3-1 所示的各种类型的激励源，允许对其参数进行设置。


表 3-1 激励源

名 称	符 号	意 义
DC		直流信号源
SINE		正弦波信号源
PULSE		模拟脉冲信号源
EXP		指数脉冲信号源
SFFM		单频率调频波信号源
PWLIN		分段线性激励源
FILE		FILE 信号源
AUDIO		音频信号源
DSTATE		数字单稳态逻辑电平信号源
DEDGE		数字单边沿信号源
DPULSE		单周期数字脉冲信号源
DCLOCK		数字时钟信号源
DPATTERN		数字模式信号源

3.1.1 直流信号源（DC Generator）

DC 信号发生器用于产生模拟直流电压或电流。这一发生器只有单一的属性：电压值或

电流值。

(1) 在 PROTEUS ISIS 环境中单击工具箱中的 Generator Mode 按钮图标，出现如图 3-1 所示的激励源列表。

(2) 单击“DC”，则在预览窗口出现直流信号源的符号，如图 3-1 所示。

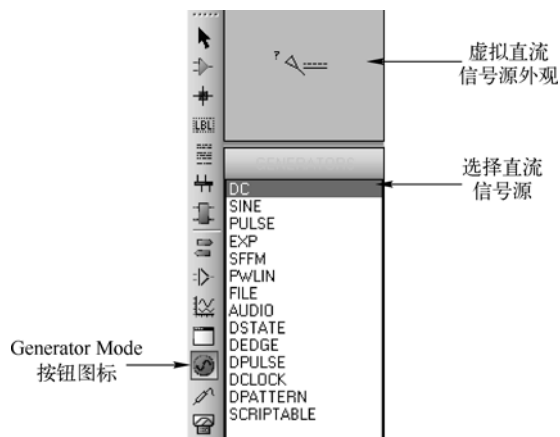


图 3-1 激励源列表

(3) 在编辑窗口双击，则直流信号源被放置到原理图编辑界面中。可使用镜像、翻转工具调整直流信号源在原理图中的位置。

(4) 在原理图编辑区中，双击直流信号源符号，出现如图 3-2 所示的属性设置对话框。

(5) 默认为直流电压源，可以在右侧设置电压源的大小。

(6) 如果需要直流电流源，则在图 3-2 中选中左侧下面的 Current Source，右侧自动出现电流值的标记，根据需要填写即可，如图 3-3 所示。

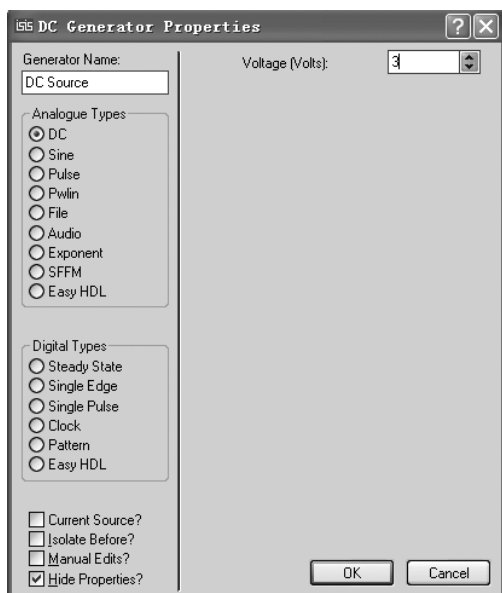


图 3-2 直流信号源的属性设置对话框（电压源）

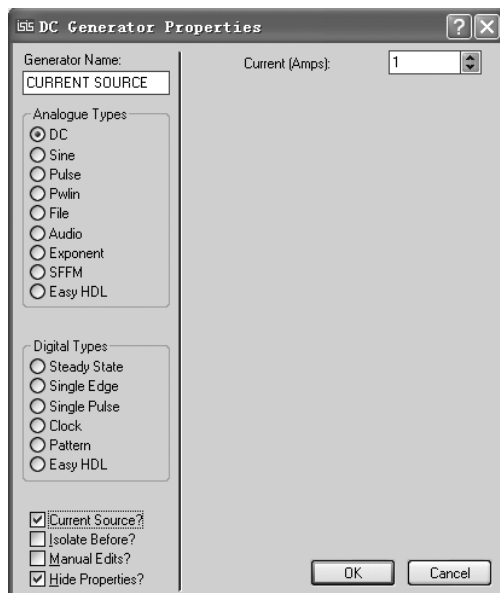



图 3-3 直流信号源的属性设置对话框（电流源）

(7) 单击“OK”按钮，完成属性设置。

(8) 用虚拟示波器观测输出。单击工具箱中的 Virtual Instrument 按钮，则在对象选择器列出所包含的项目，如图 3-4 所示。

(9) 选择 OSCILLOSCOPE 后，在编辑窗口单击，可将示波器添加到编辑窗口。

(10) 将直流信号源与示波器相连，如图 3-5 所示。

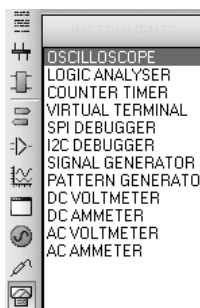


图 3-4 Virtual Instrument 图标

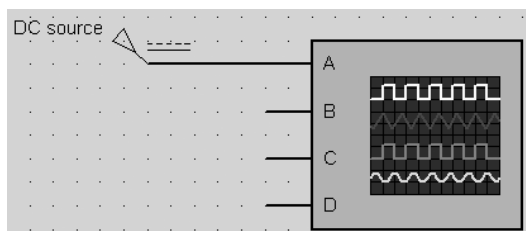



图 3-5 直流信号源与示波器连接图

(11) 单击运行按钮，出现如图 3-6 所示的仿真波形（直流源的电压值为 3V）。

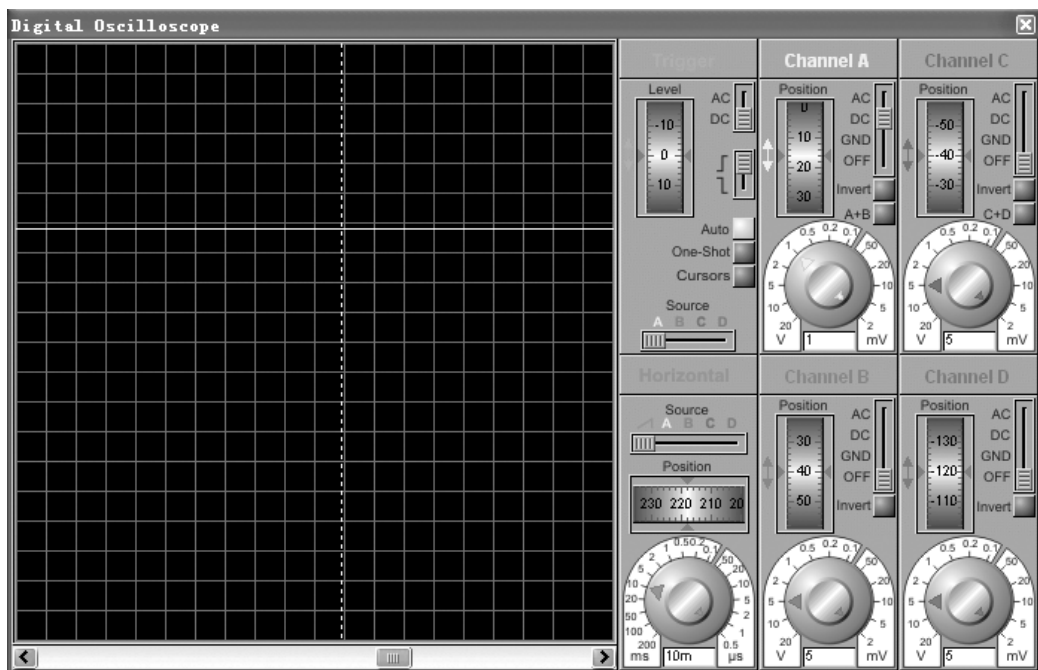



图 3-6 DC 信号源输出曲线

用户可以直接将信号发生器放置到已存在的线上，或将其放置在空白处后再连接。

3.1.2 正弦波信号源 (SINE Generator)

正弦波信号源用来产生固定频率的连续正弦波。

(1) 在 PROTEUS ISIS 环境中单击工具箱中的 Generator Mode 按钮图标，出现如图 3-1 所示的激励源列表。

- (2) 单击“SINE”，则在预览窗口出现正弦波信号源的符号。
- (3) 在编辑窗口双击，则正弦波信号源被放置到原理图编辑界面中。可使用镜像、翻转工具调整正弦波信号源在原理图中的位置。
- (4) 在原理图编辑区中，双击正弦波信号源符号，出现如图 3-7 所示的属性设置对话框。正弦波信号源属性设置对话框的主要选项含义如下。
 - ☉ Offset (Volts): 补偿电压，即正弦波的振荡中心电平。
 - ☉ Amplitude (Volts): 正弦波频率的三种定义方法，其中 Amplitude 为振幅，即半波峰值电压，Peak 为峰值电压，RMS 为有效值电压，以上三个电压值选一项即可。
 - ☉ Timing: 正弦波频率的三种定义方法，其中 Frequency (Hz)，单位为赫兹；Period (Secs) 为周期，单位为秒；这两项选一项即可。Cycles/Graph 为占空比，要单独设置。
 - ☉ Delay: 延时，指正弦波的相位，有两个选项，选填一个即可。其中 Time Delay (Secs) 是时间轴的延时，单位为秒；Phase (Degrees) 为相位，单位为度。
- (5) 在 Generator Name 文本框中输入正弦波信号源的名称，比如 SINE SOURCE 1，在相应的项目中设置相应的值。
- (6) 单击“OK”按钮，完成设置。
- (7) 连接两个正弦波信号源到示波器上，在 Generator Name 文本框中输入发生器的名称，并在相应的项目中设置相应的值。本例中使用两个正弦波信号源，其指标如表 3-2 所示。

表 3-2 两个正弦波信号源指标

信号源名称	补偿电压/V	幅值/V	频率/kHz	相位/(°)
Sine Source 1	0	1	1	0
Sine Source 2	2	2	2	90

- (8) 设置完成后，单击“OK”按钮。此时信号输入源编辑完成。
- (9) 用示波器观测输出，连接电路如图 3-8 所示。

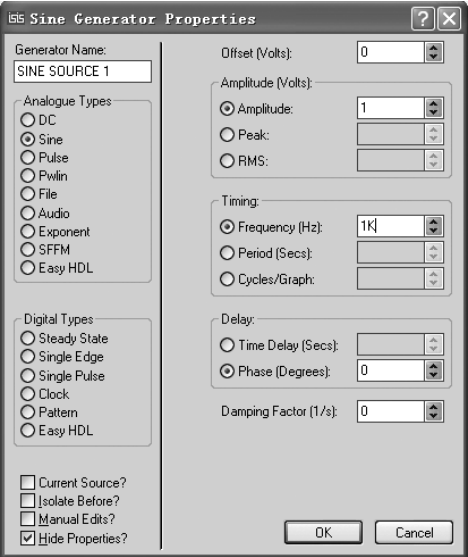


图 3-7 正弦波信号源的属性设置对话框

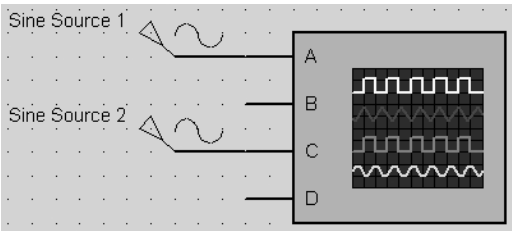


图 3-8 正弦信号源与示波器连接图

(10) 单击运行按钮，观察到示波器输出曲线如图 3-9 所示。

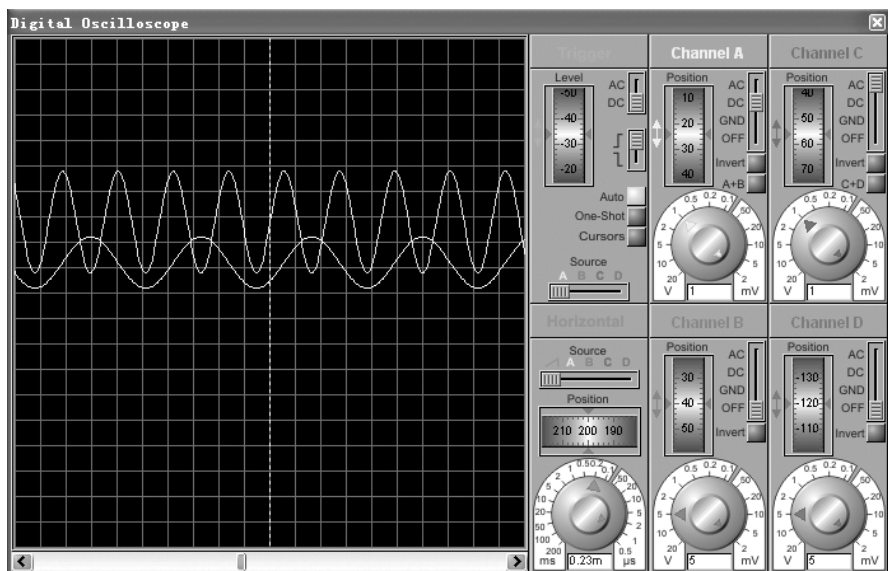



图 3-9 SINE 信号源输出曲线

3.1.3 模拟脉冲信号源 (PULSE Generator)

模拟脉冲信号源能产生各种周期的输入信号，如方波、锯齿波、三角波及单周期短脉冲等。

(1) 在 PROTEUS ISIS 环境中单击工具箱中的 Generator Mode 按钮图标，出现如图 3-1 所示的激励源列表。

(2) 单击“PULSE”，则在预览窗口出现模拟脉冲信号源的符号。

(3) 在编辑窗口双击，则模拟脉冲信号源被放置到原理图编辑界面中。可使用镜像、翻转工具调整模拟脉冲信号源在原理图中的位置。

(4) 在原理图编辑区中，双击模拟脉冲信号源符号，出现属性设置对话框，如图 3-10 所示。

- ☺ Initial (Low) Voltage: 初始 (低) 电压值。
 - ☺ Pulsed (High) Voltage: 脉冲 (高) 电压值。
 - ☺ Start (Secs): 起始时间。
 - ☺ Rise Time (Secs): 上升时间。
 - ☺ Fall Time (Secs): 下降时间。
 - ☺ Pulse Width: 脉冲宽度。有两种设置办法, Pulse Width (Secs) 指定脉冲宽度; Pulse Width (%) 指定占空比。
 - ☺ Frequency/Period: 频率或周期。
 - ☺ Current Source: 模拟脉冲信号源的电流值设置。
- (5) 设置完成后，单击“OK”按钮。

(6) 在 Generator Name 一栏中输入发生器的名称，并在相应的项目中设置合适的值。本例中 Pulse 信号源指标如表 3-3 所示。

表 3-3 Pulse 信号源指标

Generator Name	Initial(Low) Voltage	Pulsed(High) Voltage	Start	Rise Time	Fall Time	Pulse Width(%)	Frequency
Pulse Source	0	2V	0	1ms	1ms	70	100Hz

(7) 设置完成后, 单击“OK”按钮。此时信号输入源编辑完成。

(8) 用示波器观测输出, 连接电路图如图 3-11 所示。

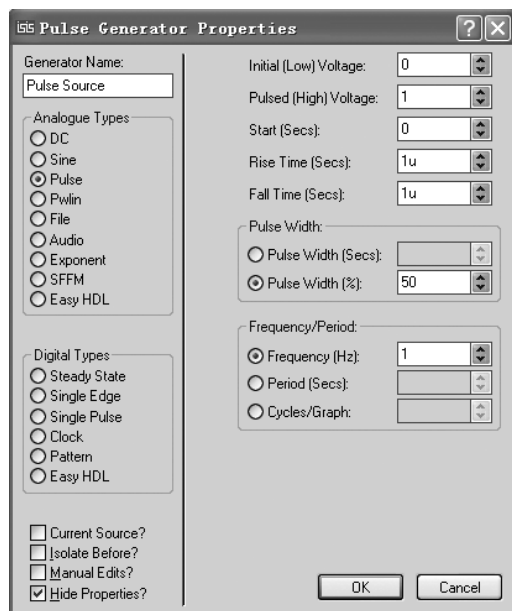


图 3-10 模拟脉冲信号源属性设置对话框

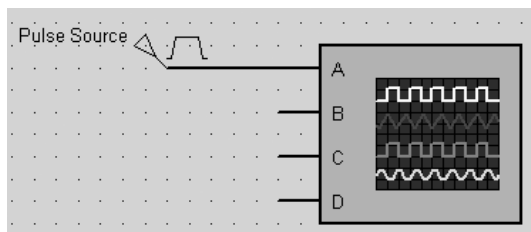


图 3-11 Pulse 信号源与示波器连接图

(9) 输出曲线如图 3-12 所示。

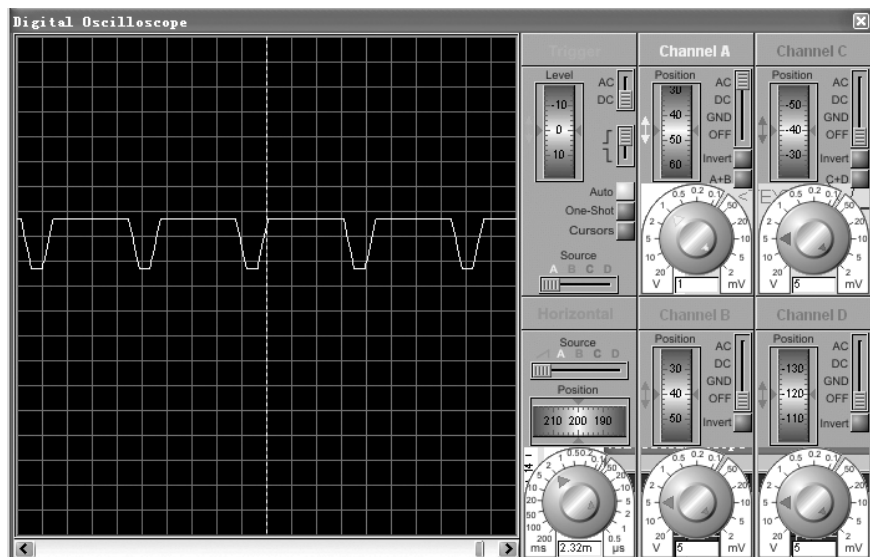



图 3-12 Pulse 信号源输出曲线

注意:

上升/下降时间不能为 0, 即不可能产生无时延的方波。

3.1.4 指数脉冲信号源 (EXP Generator)

指数脉冲信号源产生指数函数的输入信号, 其参数可以通过属性对话框来设置。

(1) 在 PROTEUS ISIS 环境中单击工具箱中的 Generator Mode 按钮图标, 出现如图 3-1 所示的激励源列表。

(2) 单击“EXP”, 则在预览窗口出现指数脉冲信号源的符号。

(3) 在编辑窗口双击, 则指数脉冲信号源被放置到原理图编辑界面中。可使用镜像、翻转工具调整指数脉冲信号源在原理图中的位置。

(4) 在原理图编辑区中, 双击指数脉冲信号源符号, 出现如图 3-13 所示的属性设置对话框。

☺ Initial (Low) Voltage: 初始 (低) 电压值。

☺ Pulsed (High) Voltage: 脉冲 (高) 电压值。

☺ Rise start time (Secs): 上升沿起始时间。

☺ Rise time constant (Secs): 上升沿持续时间。

☺ Fall start time (Secs): 下降沿起始时间。

☺ Fall time constant (Secs): 下降沿持续时间。

(5) 在图 3-13 中的 Generator Name 文本框中输入指数脉冲信号源的名称, 并在相应的项目中输入合适的值。


(6) 设置完成后, 单击“OK”按钮。

(7) 在 Generator Name 一栏中输入发生器的名称, 并在相应的项目中设置合适的值。本例中 EXP 信号源指标如表 3-4 所示。


表 3-4 EXP 信号源指标

Generator Name	Initial (Low) Voltage	Initial (High) Voltage	Rise start time (Secs)	Rise time constant (Secs)	Fall start time (Secs)	Fall time constant (Secs)
EXP Source	0	1V	0.5s	0.5s	3s	1s

(8) 设置完成后, 单击“OK”按钮。此时信号输入源编辑完成。

(9) 用模拟图表观测输出。单击工具箱中的 Simulation Graph 按钮, 在对象选择器中将出现各种仿真分析所需的图表 (如模拟、数字、噪声、混合、AC 变换等), 如图 3-14 所示。

(10) 选择 ANALOGUE 仿真图形, 并在编辑窗口单击鼠标左键, 即可将图表添加到窗口。选中图表, 单击鼠标左键, 将弹出如图 3-15 所示的对话框。在图表中设置 Stop time 值, 即图标仿真的结束时间, 本例中设置 Stop time 为 7s。

(11) 单击工具箱中的 Inter-sheet Terminal 按钮, 在对象选择器中将出现各种终端, 如图 3-16 所示。

(12) 点选 DEFAULT 选项, 然后在编辑窗口单击鼠标左键, 即可将默认终端添加到窗口。

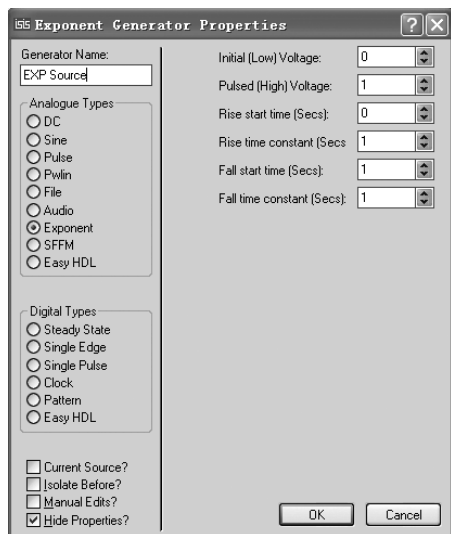


图 3-13 指数脉冲信号源属性设置对话框

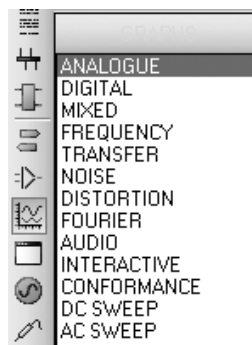


图 3-14 ISIS 编辑窗口的 Simulation Graph 工具箱

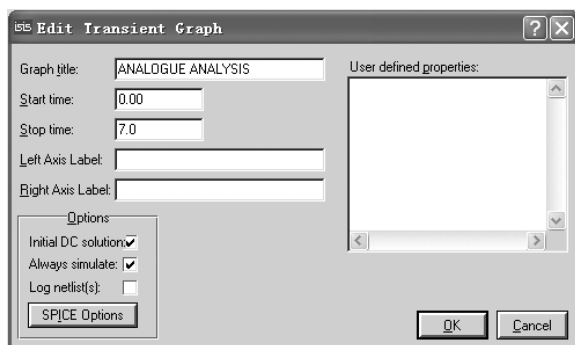


图 3-15 模拟编辑图表对话框

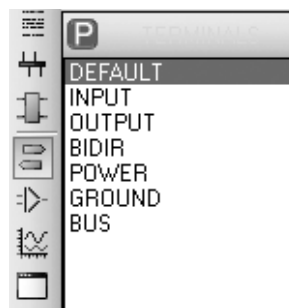


图 3-16 ISIS 编辑窗口的 Inter-sheet Terminal 工具箱

(13) 点选 EXP Source 指针，并拖动到图表中。连接电路图如图 3-17 所示。

(14) 在模拟图表上单击鼠标左键，然后单击“Space”按钮，模拟图表进行仿真。模拟图表的输出曲线如图 3-18 所示。

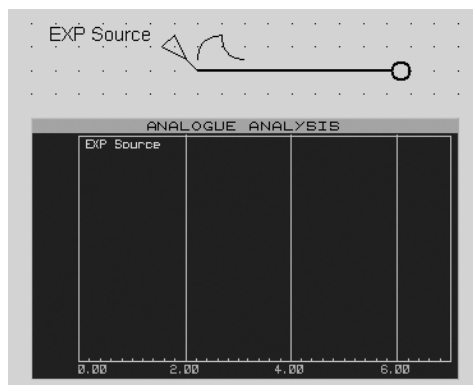


图 3-17 EXP 信号源与模拟图表连接图

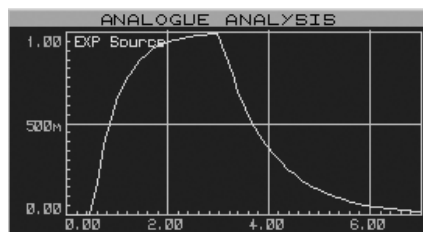



图 3-18 EXP 信号源输出曲线

3.1.5 单频率调频波信号源 (SFFM Generator)

(1) 在 PROTEUS ISIS 环境中单击工具箱中的 Generator Mode 按钮图标, 出现如图 3-1 所示的激励源列表。

(2) 单击“SFFM”, 则在预览窗口出现单频率调频波信号源的符号。

(3) 在编辑窗口双击, 则单频率调频波信号源被放置到原理图编辑界面中。可使用镜像、翻转工具调整单频率调频波信号源在原理图中的位置。

(4) 在原理图编辑区中, 双击单频率调频波信号源符号, 出现如图 3-19 所示的属性设置对话框。

☺ Offset (Volts): 电压偏置值。

☺ Amplitude (Volts): 电压幅值。

☺ Carrier Freq. (Hz): 载波频率 f_C 。

☺ Modulation Index: 调制指数 M_{DI} 。

☺ Signal Freq. (Hz): 信号频率 f_S 。

经调制后, 输出信号为 $V = V_O + V_A \sin[2\pi f_C t + M_{DI} \sin(2\pi f_S t)]$ 。

(5) 在图 3-19 中的 Generator Name 文本框中输入单频率调频波信号源的名称, 并在相应的项目中输入合适的值。

(6) 设置完成后, 单击“OK”按钮。

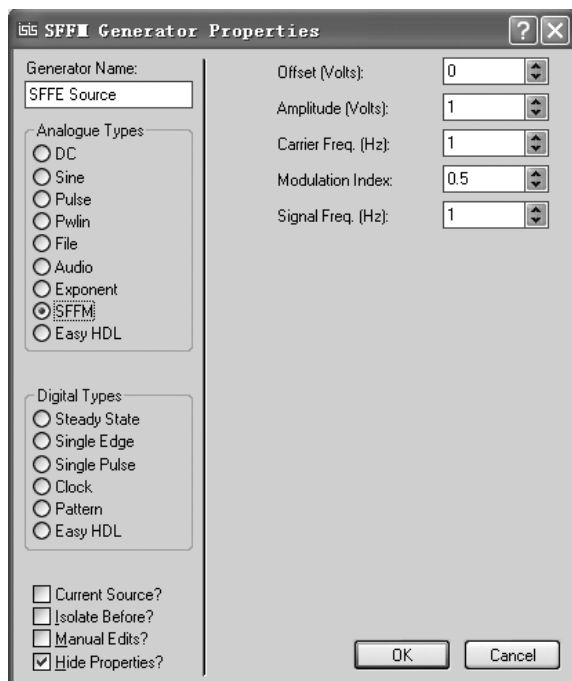


图 3-19 单频率调频波信号源属性设置对话框

(7) 在 Generator Name 一栏中输入发生器的名称, 并在相应的项目中设置合适的值。本例中 SFFM 信号源指标如表 3-5 所示。

(8) 设置完成后, 单击“OK”按钮。此时信号输入源编辑完成。

表 3-5 SFFM 信号源指标

Generator Name	V_O	V_A	f_C	M_{DI}	f_S
SFFM Source	0	1V	1Hz	0.5s	1Hz

(9) 用模拟图表观测输出。点选 SFFM Source 指针，并拖动到图表中。本例中设置图表的 Stop time 为 3s。电路图如图 3-20 所示。

(10) 在模拟图表上单击鼠标左键，然后单击“Space”按钮，模拟图表进行仿真。模拟图表输出曲线如图 3-21 所示。

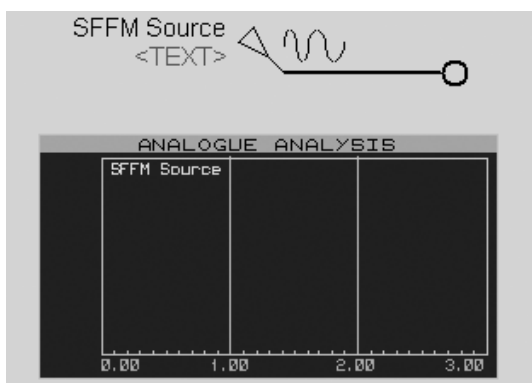


图 3-20 SFFM 信号源与模拟图表连接图

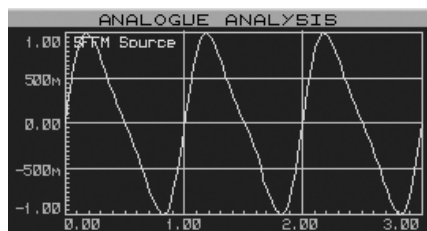



图 3-21 SFFM 信号源输出曲线

3.1.6 分段线性激励源 (PWLIN Generator)

(1) 在 PROTEUS ISIS 环境中单击工具箱中的 Generator Mode 按钮图标，出现如图 3-1 所示的激励源列表。

(2) 单击“PWLIN”，则在预览窗口出现分段线性激励源的符号。

(3) 在编辑窗口双击，则分段线性激励源被放置到原理图编辑界面中。可使用镜像、翻转工具调整分段线性激励源在原理图中的位置。

(4) 在原理图编辑区中，双击分段线性激励源符号，出现如图 3-22 所示的属性设置对话框。其中，主要参数说明如下：

☺ Time/Voltages 项：用于显示波形，X 轴为时间轴，Y 轴为电压轴。单击右上角的三角形按钮，可弹出放大的曲线编辑界面。

☺ Scaling 项：

☞ X Min: 横坐标 (时间) 最小值显示。

☞ X Max: 横坐标 (时间) 最大值显示。

☞ Y Min: 纵坐标 (时间) 最小值显示。

☞ Y Max: 纵坐标 (时间) 最大值显示。

☞ Minimum rise/fall time (Secs): 最小上升/下降时间。

(5) 用鼠标选中 Pwlin 信号发生器，单击鼠标左键，进入发生器编辑窗口，如图 3-22 所示。

(6) 在打开的分段线性激励源的图形编辑区中，用鼠标左键在任意点单击，则完成从

原点该点的一点直线，再把鼠标向右移动，在任意位置单击，又出现一连接的直线段，可编辑为自己满意的分段线性激励源曲线，如图 3-23 所示。

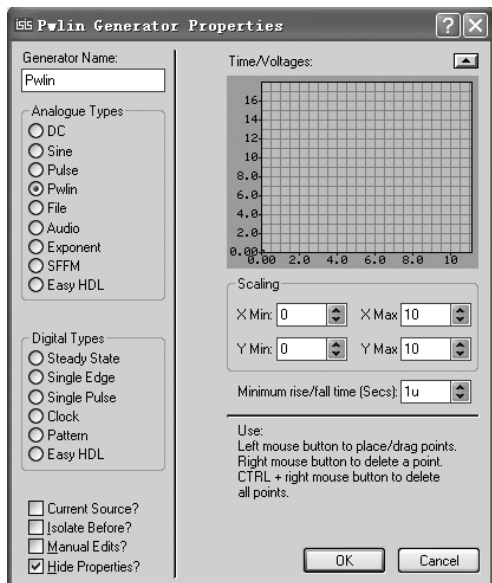


图 3-22 Pwlin 信号发生器编辑窗口

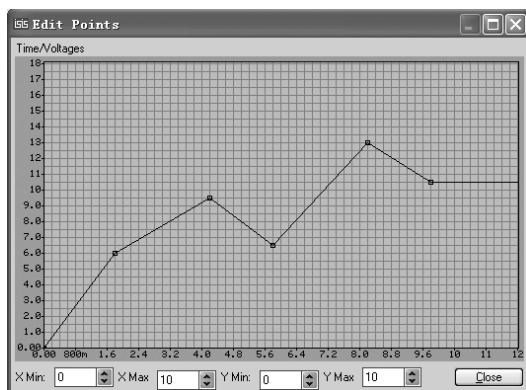


图 3-23 分段线性激励源的任意图形编辑

(7) 编辑完成后，单击“Close”按钮，关闭曲线编辑器。然后单击“OK”按钮。此时信号输入源编辑完成。

(8) 用模拟图表观测输出。点选 Pwlin Source 指针，并拖动到图表中。本例中设置图表的 Stop time 为 24s。电路图如图 3-24 所示。

(9) 在模拟图表上单击鼠标左键，然后单击“Space”按钮，模拟图表进行仿真。模拟图表输出曲线如图 3-25 所示。

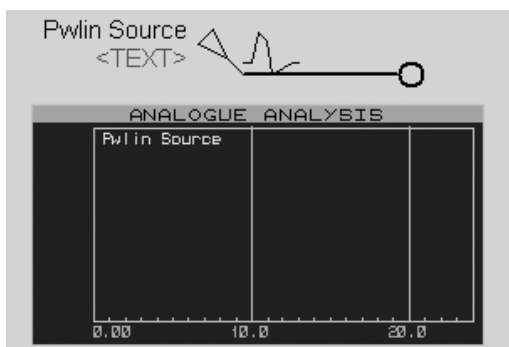


图 3-24 Pwlin 信号源与模拟图表连接图

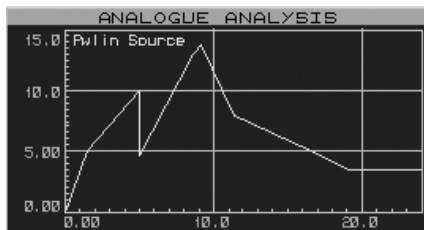


图 3-25 Pwlin 发生器输出曲线

如果用户试图编辑一个垂直边沿，则 Pwlin 曲线编辑器将按照 minimum 设定值分离垂直边沿上的两个点。

3.1.7 FILE 信号源 (FILE Generator)


(1) 在 PROTEUS ISIS 环境中单击工具箱中的 Generator Mode 按钮图标，出现如

图 3-1 所示的激励源列表。

(2) 单击“FILE”，则在预览窗口出现 FILE 信号源的符号。

(3) 在编辑窗口双击，则 FILE 信号源被放置到原理图编辑界面中。可使用镜像、翻转工具调整 FILE 信号源在原理图中的位置。

(4) 在原理图编辑区中，双击 FILE 信号源符号，出现如图 3-26 所示的属性设置对话框。在 Data File 列表中输入数据文件的路径及文件名，或单击“Browse”按钮进行路径及文件名选择，即可使用电路中编制好的数据文件。

FILE 信号源与 PWLIN 信号源相同，只是数据由 ASCII 文件产生。

ASCII 数据文件的每一行为一个时间、电压数据对，时间、电压数据对间用空格分开（也可使用 Tab 键）。时间值按升序排列，所有的值都为单精度浮点数。

(5) 在 Generator Name 文本框中输入 FILE 信号源的名称，如 FILE SOURCE。

(6) 编辑完成后，单击“OK”按钮，完成信号源的设置。

下面以一个例子来说明此种文件的格式：以下为三个周期的锯齿波，上升时间为 0.9ms，下降时间为 0.1ms，振幅为 1V。使用 UltraEdit 软件编辑 File Source.bak 文件。文件如图 3-27 所示。

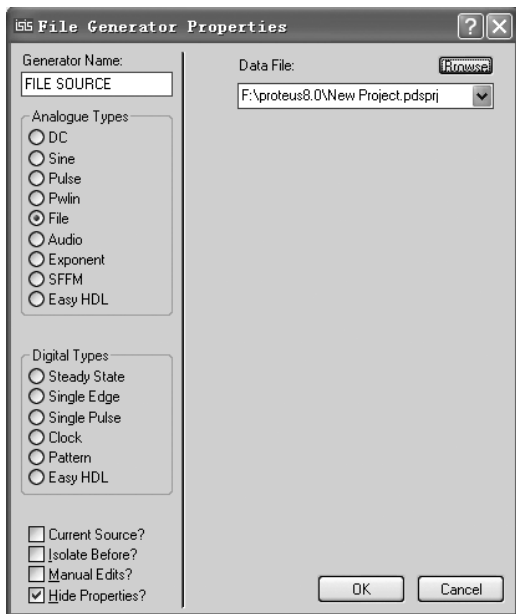


图 3-26 FILE 信号源的属性设置对话框



图 3-27 File Source.bak 文件

文件的完整数据如下：

0	0
0.9E-3	1
1E-3	0
1.9E-3	1
2E-3	0
2.9E-3	1
3E-3	0

(7) 在 Generator Name 一栏中输入发生器的名称 File Source, 并在 Data File 栏输入数据文件路径及名称。

(8) 编辑完成后, 单击“OK”按钮。此时信号输入源编辑完成。

(9) 用模拟图表观测输出。点选 File Source 指针, 并拖动到图表中。本例中设置图表的 Stop time 为 4ms。电路图如图 3-28 所示。

(10) 在模拟图表上单击鼠标左键, 然后单击“Space”按钮, 模拟图表进行仿真。模拟图表输出曲线如图 3-29 所示。

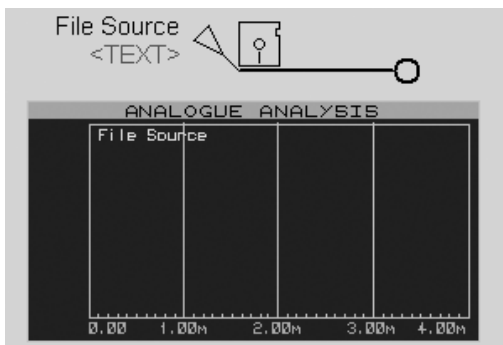


图 3-28 File 信号源与模拟图表连接图

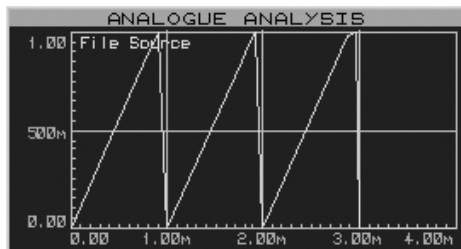



图 3-29 File 发生器输出曲线

3.1.8 音频信号源 (AUDIO Generator)

(1) 在 PROTEUS ISIS 环境中单击工具箱中的 Generator Mode 按钮图标 , 出现如图 3-1 所示的激励源列表。

(2) 单击“AUDIO”, 则在预览窗口出现音频信号源的符号。

(3) 在编辑窗口双击, 则音频信号源被放置到原理图编辑界面中。可使用镜像、翻转工具调整音频信号源在原理图中的位置。

(4) 在原理图编辑区中, 双击音频信号源符号, 出现如图 3-30 所示的音频信号源属性设置对话框。


(5) 在 Generator Name 文本框中输入音频信号源的名称, 如 Audio Source, 在 WAV Audio File 列表中, 通过单击“Browse”按钮找到一个*.wav 音频文件, 如 F:\Proteus8.0\Alarm_Sound.wav, 加载进去。

(6) 编辑完成后, 单击“OK”按钮, 完成信号源的设置。

音频文件的默认扩展名为 wav, 并且应与待分析电路在同一路径下。若不在同一路径, 须指定路径。

(7) 在 Generator Name 一栏中输入发生器的名称 Audio Source, 并在 WAV Audio File 一栏中输入目标文件的路径及其文件名 Alarm_Sound.wav。Alarm_Sound.wav 的部分内容如图 3-31 所示。

(8) 使用系统的默认设置。单击“OK”按钮。此时信号输入源编辑完成。

(9) 用音频图表观测输出。单击工具箱中的 Simulation Graph 按钮 , 在对象选择器中将出现各种仿真分析所需的图表, 如图 3-14 所示。

(10) 选择 AUDIO 仿真图形, 并在编辑窗口单击鼠标左键, 即可将图表添加到窗口。

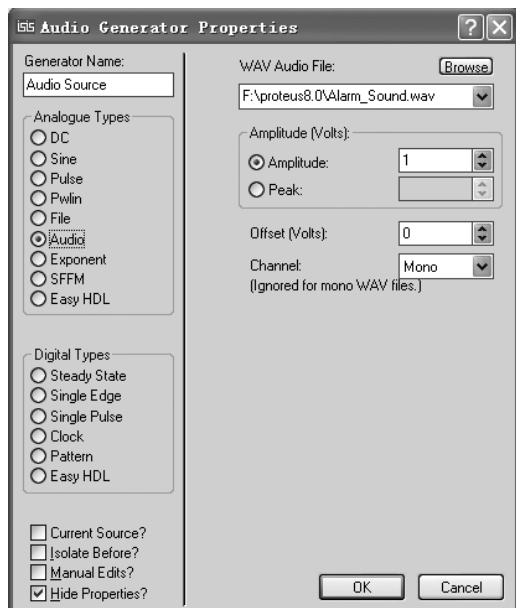


图 3-30 音频信号源属性设置对话框

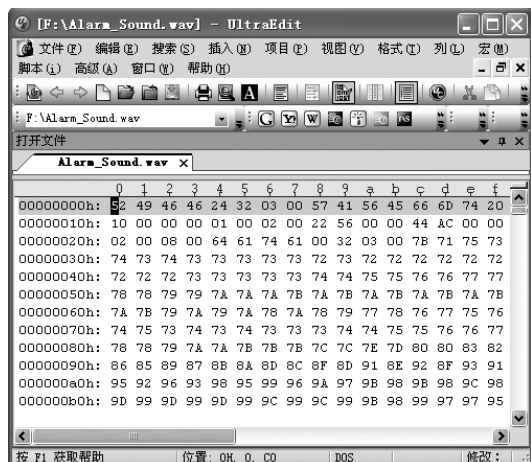


图 3-31 Alarm_Sound.wav 的部分内容

(11) 点选 Audio Source 指针，并拖动到图表中。本例中设置图表的 Stop time 为 1s。电路图如图 3-32 所示。

(12) 在音频图表上单击鼠标左键，然后单击“Space”按钮，音频图表进行仿真。音频图表输出曲线如图 3-33 所示。

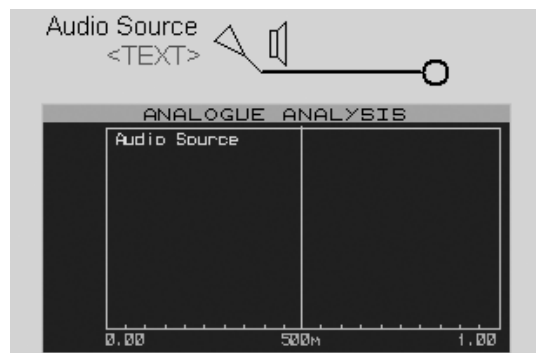


图 3-32 Audio 信号源与音频图表连接图

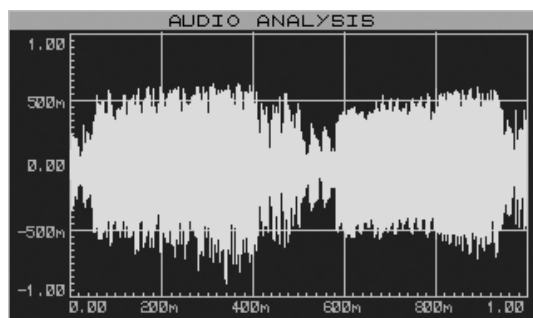



图 3-33 Audio 发生器输出曲线

3.1.9 数字单稳态逻辑电平信号源 (DSTATE Generator)

(1) 在 PROTEUS ISIS 环境中单击工具箱中的 Generator Mode 按钮图标，出现如图 3-1 所示的激励源列表。

(2) 单击“DSTATE”，则在预览窗口出现数字单稳态逻辑电平信号源的符号。

(3) 在编辑窗口双击，则数字单稳态逻辑电平信号源被放置到原理图编辑界面中。可使用镜像、翻转工具调整数字单稳态逻辑电平信号源在原理图中的位置。

(4) 在原理图编辑区中，双击数字单稳态逻辑电平信号源，出现如图 3-34 所示的数字

单稳态逻辑电平信号源属性设置对话框。

(5) 在 Generator Name 文本框中输入数字单稳态逻辑电平信号源的名称, 如 DState Source 1, 在 State 选项组中, 逻辑状态为 Weak Low (弱低电平)。

(6) 编辑完成后, 单击“OK”按钮, 完成信号源的设置。

(7) 在 Generator Name 一栏中输入发生器的名称, 并在相应的项目中设置合适的值。本例中使用七个 DState 信号发生器, 来查看它们的输出。七个 DState 信号发生器的设置如表 3-6 所示。

表 3-6 数字单稳态逻辑电平信号源指标

Generator Name	State
DState Source 1	Power Rail High
DState Source 2	Strong High
DState Source 3	Weak High
DState Source 4	Floating
DState Source 5	Weak Low
DState Source 6	Strong Low
DState Source 7	Power Rail Low

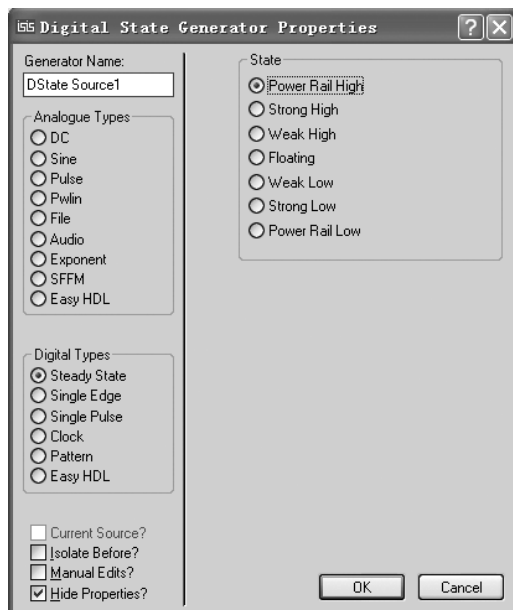


图 3-34 数字单稳态逻辑电平信号源属性设置对话框

(8) 设置完成后, 单击“OK”按钮。此时信号输入源编辑完成。

(9) 分别用模拟图表和数字图表观测输出。点选 DState Source 1~DState Source 7 指针, 并拖动到图表中。本例中设置图表的 Stop time 为 1s。电路图如图 3-35 所示。

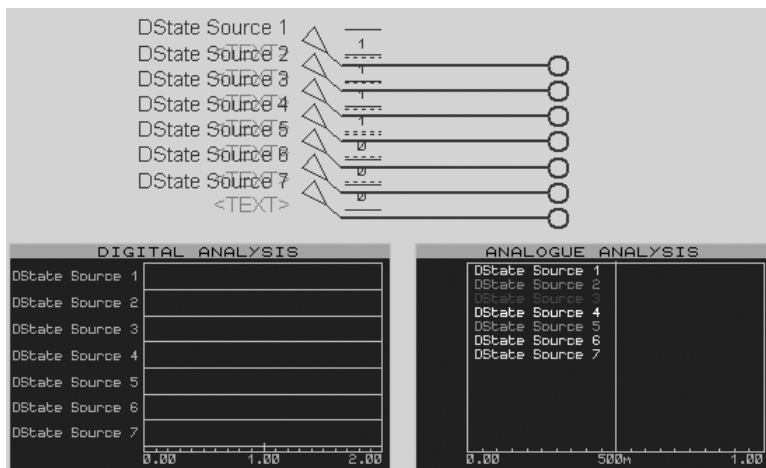


图 3-35 DState 信号源与数字、模拟图表连接图

(10) 分别在数字图表、模拟图表上单击鼠标左键，然后单击“Space”按钮，数字图表、模拟图表进行仿真。模拟图表、数字图表输出曲线如图 3-36 所示。其中：Power Rail High、Strong High、Weak High 重合；Weak Low、Strong Low、Power Rail Low 重合。

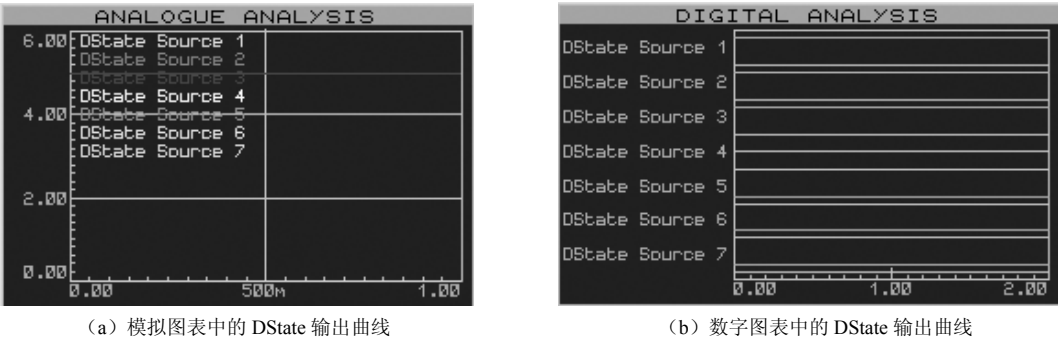



图 3-36 模拟图表、数字图表输出曲线

3.1.10 数字单边沿信号源（DEEDGE Generator）

(1) 在 PROTEUS ISIS 环境中单击工具箱中的 Generator Mode 按钮图标，出现如图 3-1 所示的激励源列表。

(2) 单击“DEEDGE”，则在预览窗口出现数字单边沿信号源的符号。

(3) 在编辑窗口双击，则数字单边沿信号源被放置到原理图编辑界面中。可使用镜像、翻转工具调整数字单边沿信号源在原理图中的位置。

(4) 在原理图编辑区中，双击数字单边沿信号源，出现如图 3-37 所示的数字单边沿信号源属性设置对话框。

(5) 在 Generator Name 文本框中输入数字单边沿信号源的名称，如 DEDGE 1，在 Edge Polarity 栏选择 Positive（Low-To-High）Edge 正边沿项。对于 Edge At（Secs）项，输入 500m，即选择边沿发生在 500ms 处。

(6) 编辑完成后，单击“OK”按钮，完成信号源的设置。

(7) 在 Generator Name 一栏中输入发生器的名称，并在相应的项目中设置合适的值。本例中 DEDGE 信号发生器指标如表 3-7 所示。

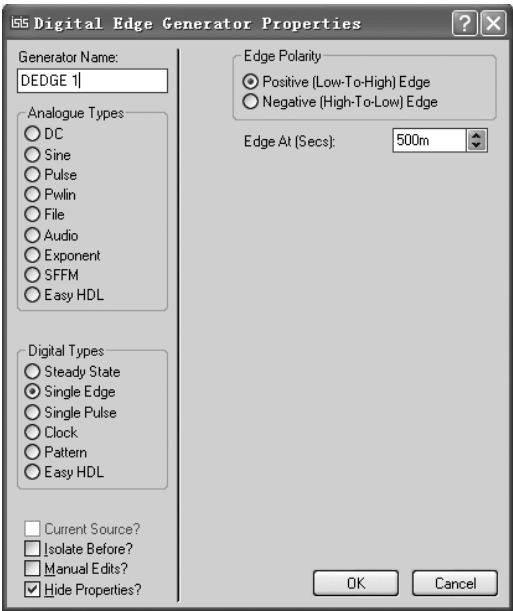


图 3-37 数字单边沿信号源属性设置对话框

表 3-7 数字单边沿信号源指标

Generator Name	Edge Polarity	Edge At(Secs)
DEEDGE Source	Positive(Low-To-High) Edge	1s

(8) 设置完成后, 单击“OK”按钮。此时信号输入源编辑完成。

(9) 用数字图表观测输出。点选 DEDGE Source 指针, 并拖动到图表中。本例中设置图表的 Stop time 为 2s。电路图如图 3-38 所示。

(10) 在数字图表上单击鼠标左键, 然后单击“Space”按钮, 数字图表进行仿真。数字图表输出曲线如图 3-39 所示。

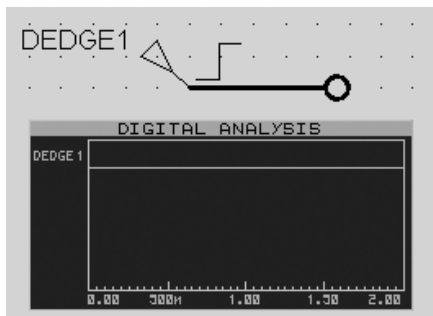


图 3-38 DEDGE 信号源与数字图表连接图

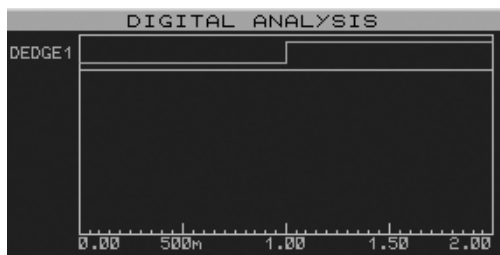



图 3-39 DEDGE 1 信号源输出曲线

3.1.11 单周期数字脉冲信号源 (DPULSE Generator)

(1) 在 PROTEUS ISIS 环境中单击工具箱中的 Generator Mode 按钮图标, 出现如图 3-1 所示的激励源列表。

(2) 单击“DPULSE”, 则在预览窗口出现单周期数字脉冲信号源的符号。

(3) 在编辑窗口双击, 则单周期数字脉冲信号源被放置到原理图编辑界面中。可使用镜像、翻转工具调整单周期数字脉冲信号源在原理图中的位置。

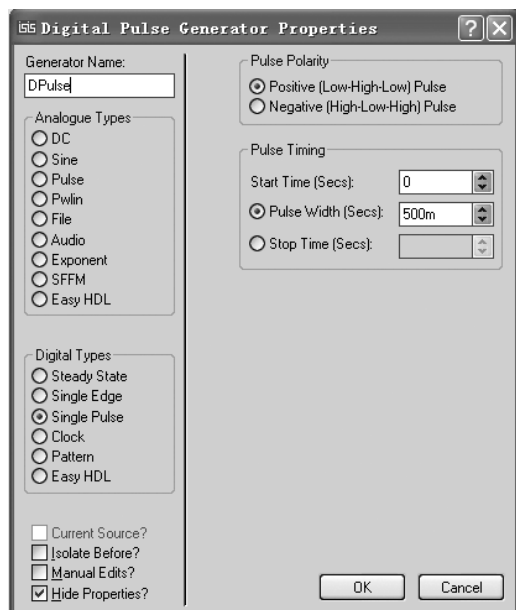


图 3-40 单周期数字脉冲信号源属性设置对话框

(4) 在原理图编辑区中, 双击单周期数字脉冲信号源, 出现如图 3-40 所示的单周期数字脉冲信号源属性设置对话框。主要有以下参数设置。

☺ Pulse Polarity (脉冲极性): 正脉冲 Positive (Low-High-Low) Pulse 和负脉冲 Negative (High-Low-High) Pulse。

☺ Pulse Timing (脉冲定时): Start Time (Secs) 为起始时刻; Pulse Width (Secs) 为脉宽; Stop Time (Secs) 为停止时间。

(5) 在 Generator Name 文本框中输入单周期数字脉冲信号源的名称, 如 DPulse Source, 并在相应的项目中设置合适的值。


(6) 编辑完成后, 单击“OK”按钮, 完成信号源的设置。

(7) 在 Generator Name 一栏中输入发生器的名称, 并在相应的项目中设置合适的值。本例中 DPulse 信号发生器指标如表 3-8 所示。

表 3-8 单周期数字脉冲信号源指标

Generator Name	Pulse Polarity	Start Time(Secs)	Pulse Width (Secs)	Stop Time (Secs)
DPulse Source	Negative (High-Low-High) Pluse	1.3 μ	1.2 μ	—

(8) 设置完成后, 单击“OK”按钮。此时信号输入源编辑完成。

(9) 用数字图表观测输出。单击工具箱中的 Simulation Graph 按钮, 在对象选择器中将出现各种仿真分析所需的图表, 如图 3-14 所示。

(10) 选择 DIGITAL 仿真图形, 并在编辑窗口单击鼠标左键, 即可将图表添加到窗口。

(11) 点选 DPulse Source 指针, 并拖动到图表中。本例中设置图表的 Stop time 为 3 μ s。电路图如图 3-41 所示。

(12) 在数字图表上单击鼠标左键, 然后单击“Space”按钮, 数字图表进行仿真。数字图表输出曲线如图 3-42 所示。

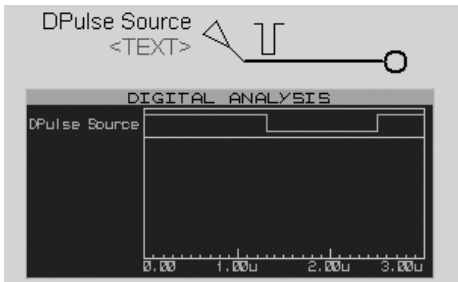


图 3-41 DPulse 信号源与数字图表连接图

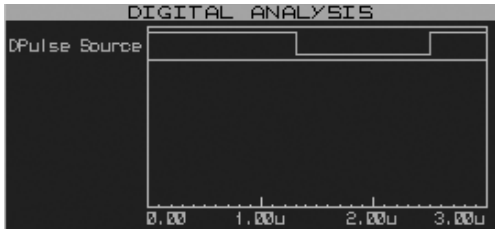


图 3-42 DPulse 信号源输出曲线

3.1.12 数字时钟信号源 (DCLOCK Generator)

(1) 在 PROTEUS ISIS 环境中单击工具箱中的 Generator Mode 按钮图标, 出现如图 3-1 所示的激励源列表。

(2) 单击“DCLOCK”, 则在预览窗口出现数字时钟信号源的符号。

(3) 在编辑窗口双击, 则数字时钟信号源被放置到原理图编辑界面中。可使用镜像、翻转工具调整数字时钟信号源在原理图中的位置。

(4) 在原理图编辑区中, 双击数字时钟信号源, 出现如图 3-43 所示的数字时钟信号源属性设置对话框。

(5) 在 Generator Name 文本框中输入数字时钟信号源的名称, 如 DClock Source, 并在 Timing 选项组中把 Frequency (Hz) 频率设为 1kHz。

(6) 编辑完成后, 单击“OK”按钮, 完成信号源的设置。

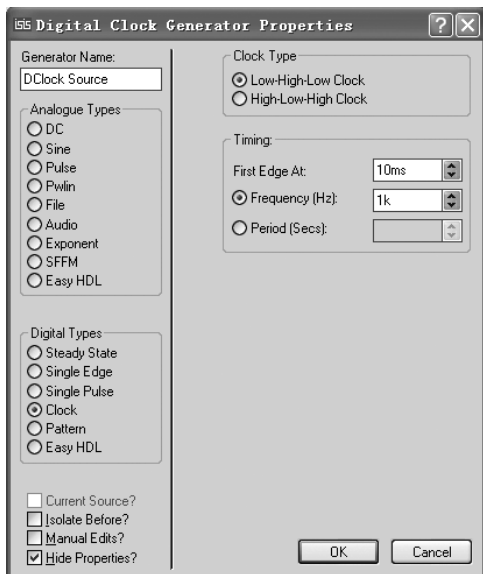


图 3-43 数字时钟信号源属性设置对话框

(7) 在 Generator Name 一栏中输入发生器的名称, 并在相应的项目中设置合适的值。本例中 DClock 信号发生器指标如表 3-9 所示。

表 3-9 DClock 信号源指标

Generator Name	Clock Type	First Edge At	Frequency (Hz)	Period(Secs)
DClock Source	Low-High-Low Clock	10ms	1kHz	—

(8) 设置完成后, 单击“OK”按钮。此时信号输入源编辑完成。

(9) 用数字图表观测输出。点选 DClock Source 指针, 并拖动到图表中。本例中设置图表的 Stop time 为 20ms。电路图如图 3-44 所示。

(10) 在数字图表上单击鼠标左键, 然后单击“Space”按钮, 数字图表进行仿真。数字图表输出曲线如图 3-45 所示。

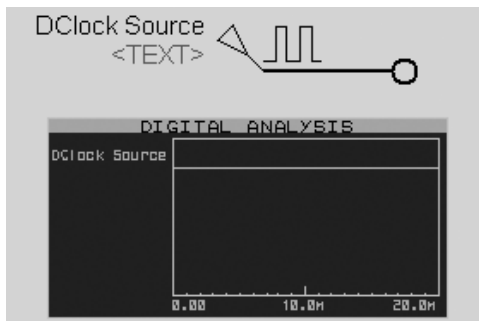


图 3-44 DClock 信号源与数字图表连接图

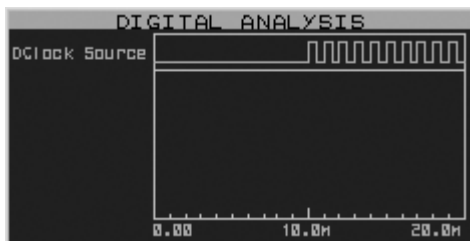


图 3-45 DClock 信号源输出曲线

3.1.13 数字模式信号源 (DPATTERN Generator)

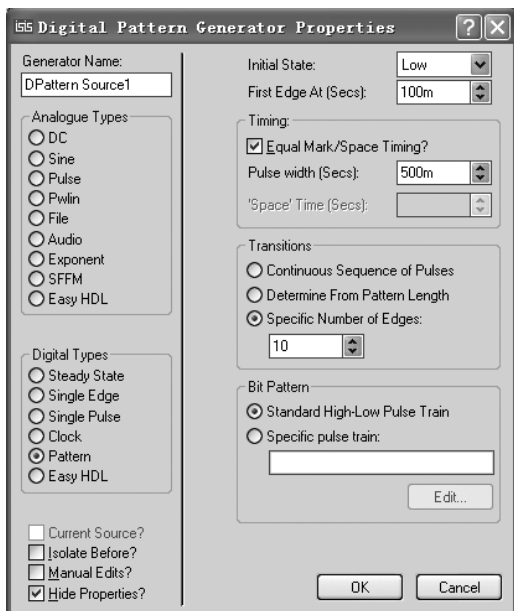



图 3-46 数字模式信号源属性设置对话框

(1) 在 PROTEUS ISIS 环境中单击工具箱中的 Generator Mode 按钮图标, 出现如图 3-1 所示的激励源列表。

(2) 单击“DPATTERN”, 则在预览窗口出现数字模式信号源的符号。

(3) 在编辑窗口双击, 则数字模式信号源被放置到原理图编辑界面中。可使用镜像、翻转工具调整数字模式信号源在原理图中的位置。

(4) 在原理图编辑区中, 双击数字模式信号源, 出现如图 3-46 所示的数字模式信号源属性设置对话框。

(5) 在 Generator Name 文本框中输入单周期数字脉冲信号源的名称, 如 DPattern Source 1, 其中各项含义如下:

☺ Initial State: 初始状态。

- ☺ First Edge At (Secs): 脉冲宽度。
- ☺ Specific Number of Edges: 指定脉冲边沿数目。
- ☺ Specific pulse train: 指定脉冲轨迹。

(6) 在指定脉冲轨迹项的下边单击“Edit”按钮，出现如图 3-47 所示的数字模式信号源的轨迹编辑区。

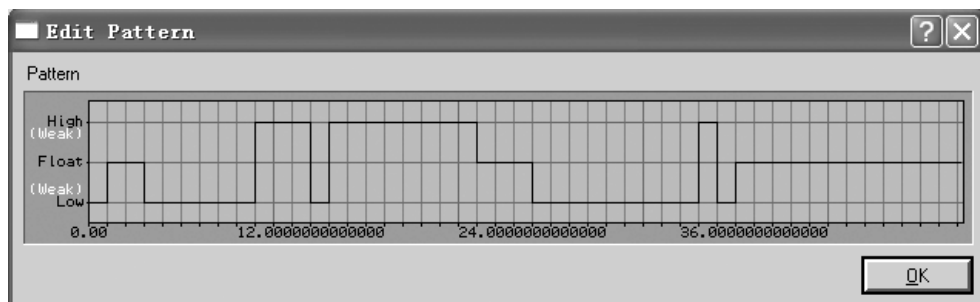


图 3-47 数字模式信号源的轨迹编辑区

(7) 在图 3-47 中，通过单击鼠标可以确定轨迹，有高电平、低电平和浮动电平三种电平可以改变。单击“OK”按钮完成轨迹编辑，返回图 3-46 所示的属性对话框。

(8) 单击“OK”按钮完成属性设置。

(9) 在 Generator Name 一栏中输入发生器的名称，并编辑信号。在本例中使用三个信号源，对它们分别进行如表 3-10 所示的编辑。

表 3-10 DPattern 信号源指标

Generator Name	Initial State	First Edge At (Secs)	Timing		Transitions		Bit Pattern	
DPattern Source 1	Low	100ms	Equal Mark/Space Timing	✓	Specific Number of Edges	10	Standard High-Low Pulse Train	
			Pulse width (Secs)	500ms				
DPattern Source 2	Low	100ms	Mark Time	300ms	Continuous Sequence of Pulses		Specific pulse train	11000HLfhl, 如图 3-48 所示
			Space Time	700ms				
DPattern Source 3	Low	100ms	Equal Mark/Space Timing	✓	Continuous Sequence of Pulses		Specific pulse train	使用 Edit 进行编辑, 如图 3-49 所示
			Pulse width (Secs)	500ms				

(10) 设置完成后，单击“OK”按钮。此时信号输入源编辑完成。

(11) 用数字图表观测输出。点选 DPattern Source 1、DPattern Source 2 和 DPattern Source 3 指针，并拖动到图表中。本例中设置图表的 Stop time 为 15s。电路图如图 3-50 所示。

(12) 在数字图表上单击鼠标左键，然后单击“Space”按钮，数字图表进行仿真。数字图表输出曲线如图 3-51 所示。

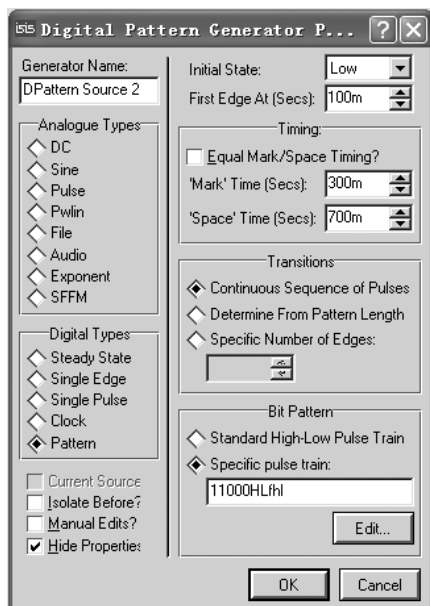


图 3-48 DPattern Source 3 编辑对话框

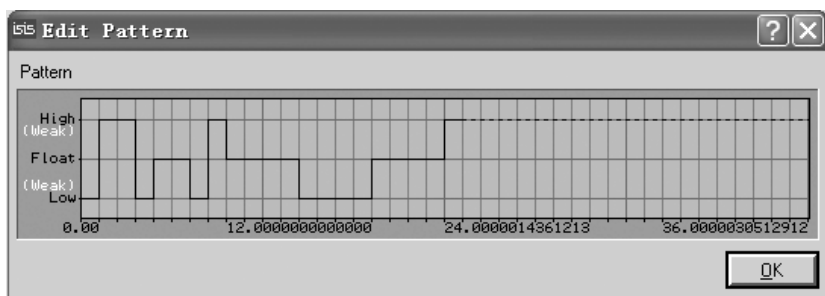


图 3-49 DPattern Source 3 曲线

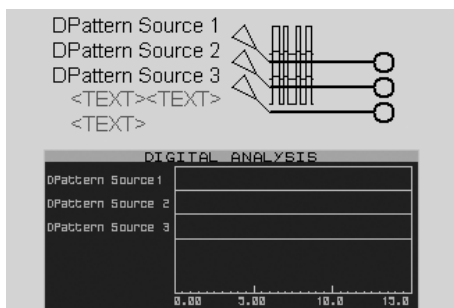


图 3-50 DPattern 信号源与数字图表连接图

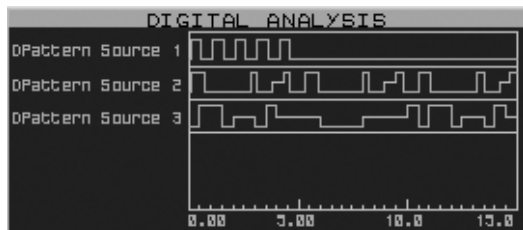


图 3-51 DPattern 信号源输出曲线




3.2 基于图表的分析

图表分析可以得到整个电路分析结果，并且可以直观地对仿真结果进行分析。同时，图表

分析能够在仿真过程中放大一些特别的部分，进行一些细节上的分析。另外，图表分析也是唯一一种能够实现在实时中难以做出的分析，比如说交流小信号分析、噪声分析和参数扫描。

图表在仿真中是一个最重要的部分。它不仅是结果的显示媒介，而且定义了仿真类型。通过放置一个或若干个图表，用户可以观测到各种数据（数字逻辑输出、电压、阻抗等），即通过放置不同的图表来显示电路在各方面的特性。

单击工具箱中的按钮，在对象拾取器中列出所有的电路分析图表如图 3-52 所示。电路分析图表及其含义见表 3-11。

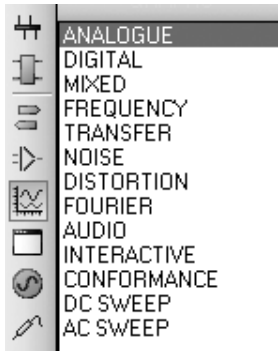


图 3-52 电路分析图表列表


表 3-11 电路分析图表及其含义

名 称	含 义
ANALOGUE	模拟分析图表
DIGITAL	数字分析图表
MIXED	混合分析图表
FREQUENCY	频率分析图表
TRANSFER	转移特性分析图表
NOISE	噪声分析图表
DISTORTION	失真分析图表
FOURIER	傅里叶分析图表
AUDIO	音频分析图表
INTERACTIVE	交互分析图表
CONFORMANCE	一致性分析图表
DC SWEEP	直流扫描分析图表
AC SWEEP	交流扫描分析图表

3.2.1 基于模拟分析图表的电路分析

模拟分析图表用于绘制一条或多条电压或电流随时间变化的曲线。

1. 放置图表

(1) 单击工具箱中的 Simulation Graph 图标，在对象选择器中将出现各种仿真分析所需的图表，选择 ANALOGUE 仿真图表。

(2) 在编辑窗口期望放置图表的位置单击鼠标左键，并拖动鼠标，此时将出现一个矩形图表轮廓，如图 3-53 所示。

(3) 在期望的结束点单击鼠标左键，放置图表，如图 3-54 所示。

图表与其他元件在移动、删除和编辑等方面的操作相同。

图表的大小可以调整：右击图表，图表被选中，四边出现小黑方框，光标指向方框拖动即可调整图表大小。

2. 放置发生器和探针

每个发生器都默认自带一个探针，所以不需要再为发生器放置探针。有三种方法可以

加入发生器和探针：

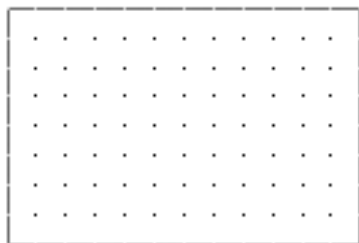


图 3-53 选择位置

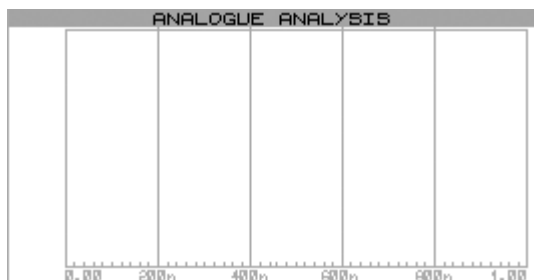


图 3-54 放置图表

(1) 依次选中探针或发生器，拖动到图表中。图表有左右两条竖轴，探针/发生器靠近哪边被拖入，其名字就被放置在那条轴上，图表中的探针/发生器与原理图的名字相同。

(2) 当原理图中没有被选中的探针/发生器时，选择 **Graph→Add Trace** 菜单项，单击“OK”按钮，则把所有选中的探针放置到图表中，以字母顺序排序。

(3) 如果原理图中有被选中的探针/发生器，那么选择 **Graph→Add Trace** 菜单项，单击“OK”按钮，则把所有选中的探针放置到图表中，以字母顺序排序。

不同的探针名称和发生器由不同的颜色表示。

同其他元件一样，右击探针名（或发生器名）选中探针名（或发生器名），探针名变为白色，拖动探针名（或发生器名）来调整顺序，也可以把左边竖轴的探针名（或发生器名）放到右边的竖轴。

3. 设置仿真图表

双击模拟图表，将弹出如图 3-55 所示的模拟图表编辑对话框。

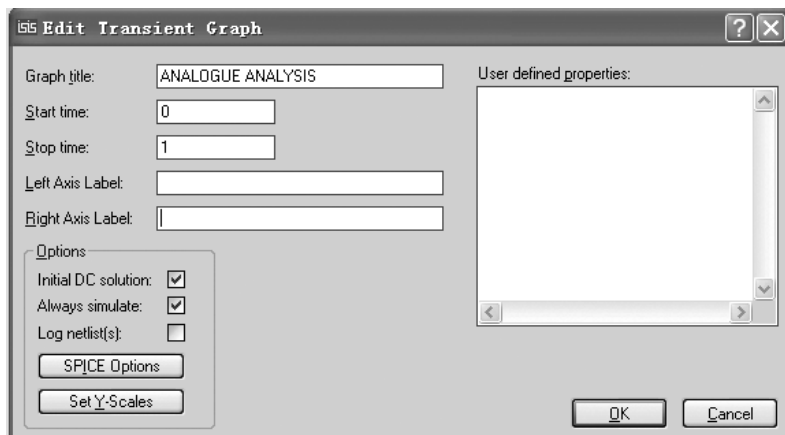


图 3-55 模拟图表编辑对话框

对话框中包含如下设置内容：

- ☺ **Graph title:** 图表标题。
- ☺ **Start time:** 仿真起始时间。
- ☺ **Stop time:** 仿真终止时间。

☺ Left Axis Label: 左边坐标轴标签。

☺ Right Axis Label: 右边坐标轴标签。

设置完成后, 单击“OK”按钮。

本例中添加的发生器和探针为 INPUT 和 OUTPUT 两信号, 设置停止时间为 1ms, 如图 3-56 所示。

4. 仿真

对图 3-57 所示电路进行仿真。

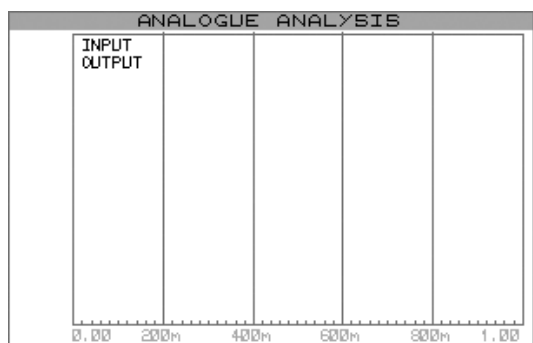


图 3-56 编辑后的图表

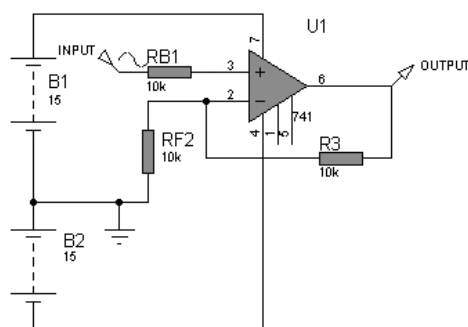


图 3-57 模拟图表仿真电路图

将鼠标指针放在图表上, 选择 Graph→Simulate (快捷键: 空格键) 即可。仿真命令使电路开始仿真, 图表也随仿真的结果进行更新。仿真日志记录最后一次的仿真情况, 用 Graph→View Log (快捷键: Ctrl+V) 可实现仿真日志的记录。当仿真中出现错误时, 日志中可显示详细的出错信息。

如果再一次执行仿真, 可看到图表并没有发生变化, 可在编辑瞬时图表对话框中选择 Always Simulate, 此时, 就可看到图表在动态刷新。当可以看到图表上的波形, 但并不能看清细节时, 用鼠标左击图表的标题栏, 可把图表最大化 (全编辑窗口显示), 如图 3-58 所示。分析完成后, 用鼠标左击图表标题栏可恢复原编辑窗口。

当显示窗口中两条曲线幅值相差太大时, 如图 3-59 所示。

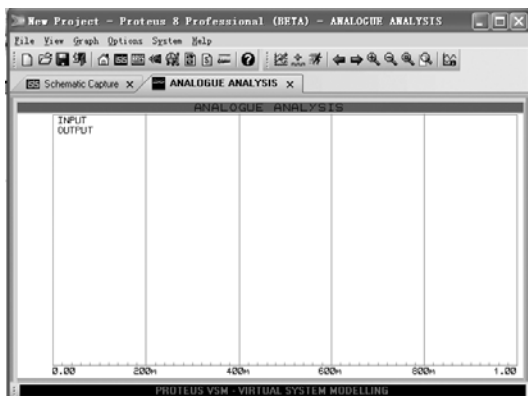


图 3-58 模拟图表以窗口形式出现

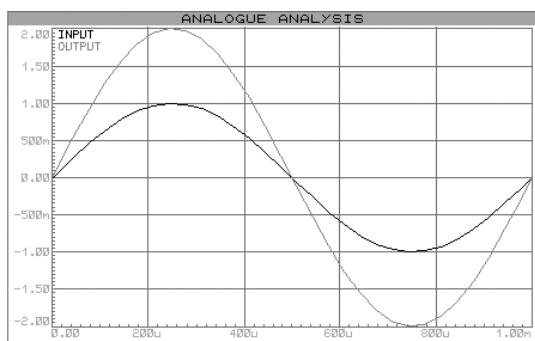



图 3-59 例图 (两条曲线幅值相差偏大时)

可以用分离的方法：选中 OUTPUT 信号，按下左键拖动到右边的竖轴，看到如图 3-60 所示的显示窗口。

 注意：

两边竖轴的单位是不同的。

测量时，需放置两条测量线（平行于竖轴）：在图表中左击鼠标，出现一条粉红色线（基本指针），按下 Ctrl 键，在图表中左击鼠标，出现另一条蓝色线（参考指针），如图 3-61 所示。

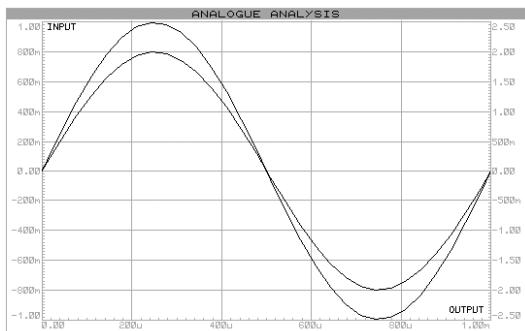


图 3-60 分离曲线

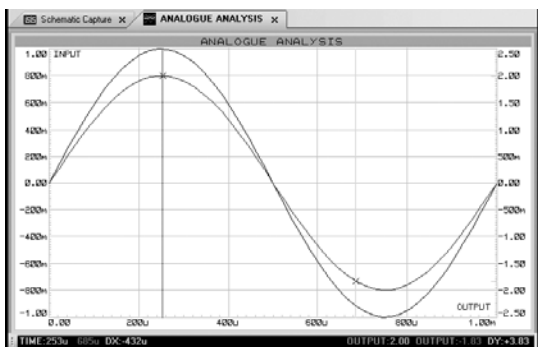


图 3-61 使用基本指针与参考指针进行测量

移动测量线时也一样：左击鼠标移动粉红色线，按下 Ctrl 键左击鼠标则移动蓝线。

删除测量线：鼠标指向任一竖轴的标值（如左轴的 -200m、400m 等）左击鼠标删除绿线；鼠标指向任一竖轴的标值，按下 Ctrl 键，左击鼠标，删除红线。

每个图表中只能出现两条测量线，对两个量进行测量。

此时图表底部为状态栏，显示的数据都是绝对值。其中 DX 显示时间相对量；DY 显示幅值相对量。

3.2.2 基于数字分析图表的电路分析

数字分析图表用于绘制逻辑电平值随时间变化的曲线，图表中的波形代表单一数据位或总线的二进制电平值。

1. 放置数字分析图表

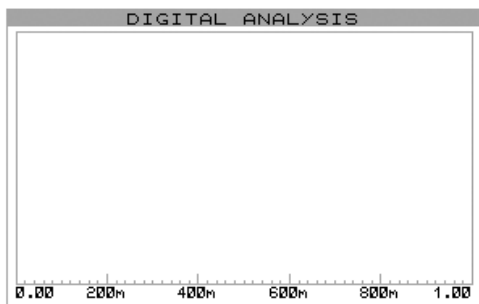



图 3-62 数字分析图表

单击工具箱中的 Simulation Graph  图标，在对象选择器中将出现各种仿真分析所需的图表。选择 DIGITAL 仿真图表，在编辑窗口期望放置图表的位置单击鼠标左键，并拖动鼠标，此时将出现一个矩形图表轮廓。在期望的结束点单击鼠标左键，放置图表，如图 3-62 所示。

2. 设置数字分析图表

双击图表将弹出如图 3-63 所示的数字分析

图表编辑对话框。

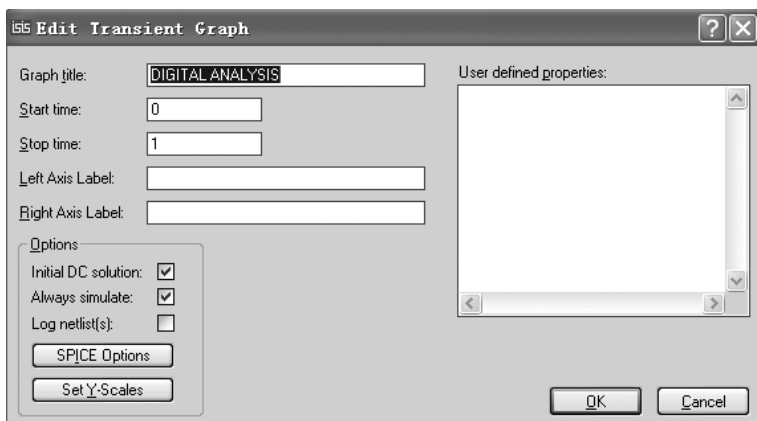


图 3-63 数字分析图表编辑对话框

对话框中包含如下设置内容：

- ☺ Graph title: 图表标题。
- ☺ Start time: 仿真起始时间。
- ☺ Stop time: 仿真终止时间。
- ☺ Left Axis Label: 左边坐标轴标签。
- ☺ Right Axis Label: 右边坐标轴标签。

3. 进行仿真

对例图 3-64 进行仿真。

将 S、R、U1: A (\bar{Q}) 和 U1: B (\bar{Q}) 添加到数字图表。运行 Graph→Simulate (快捷键: 空格键) 仿真命令, 即可开始仿真。数字图表的输出结果如图 3-65 所示。

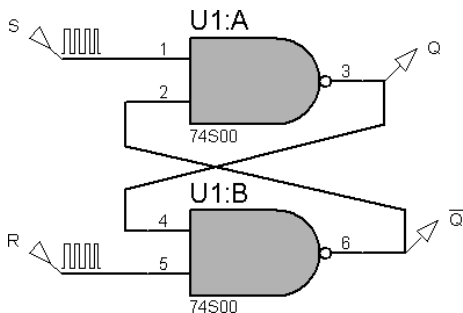


图 3-64 数字图表仿真例图

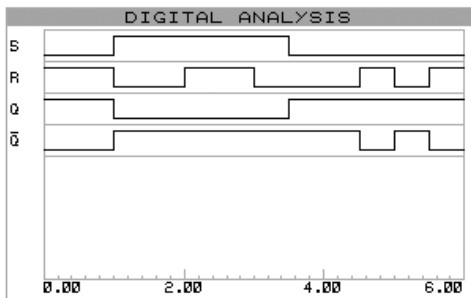



图 3-65 数字图表输出结果

3.2.3 基于混合分析图表的电路分析

混合分析图表可以在同一图表中同时显示模拟和数字信号的波形。

1. 放置混合分析图表

单击工具箱中的 Simulation Graph 图标 , 在对象选择器中选择 MIXED 仿真图表。在

编辑窗口期望放置图表的位置单击鼠标左键，并拖动鼠标，在期望的结束点单击鼠标左键，放置混合分析图表，如图 3-66 所示。

2. 设置混合分析图表

双击图表将弹出如图 3-67 所示的混合分析图表属性编辑对话框。

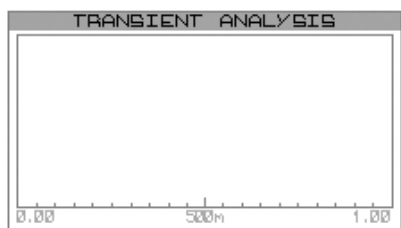


图 3-66 混合分析图表

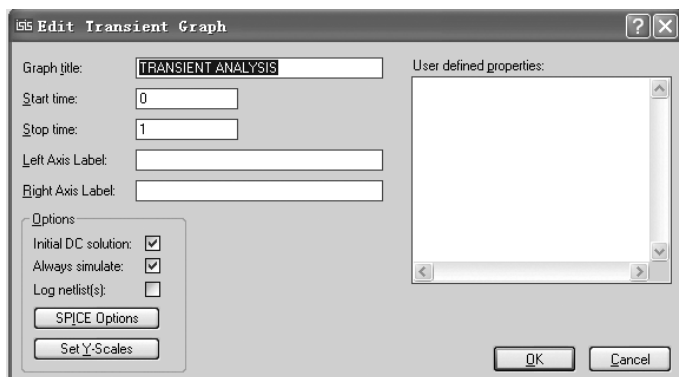


图 3-67 混合分析图表属性编辑对话框

对话框中包含如下设置内容：

- ☺ Graph title: 图表标题。
 - ☺ Start time: 起始仿真时间。
 - ☺ Stop time: 终止仿真时间。
 - ☺ Left Axis Label: 左边坐标轴标签。
 - ☺ Right Axis Label: 右边坐标轴标签。
- 设置完成后，单击“OK”按钮，结束设置。

3. 进行仿真

本例中使用图 3-68 所示电路进行仿真。

(1) 添加探针和发生器。添加第一个数字探针时，鼠标放置在图标上，单击右键，选择 Add Trace 命令，出现如图 3-69 所示的对话框。

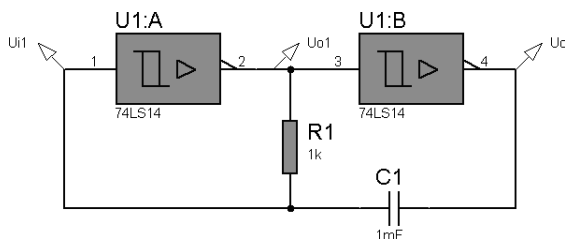


图 3-68 混合电路仿真例图

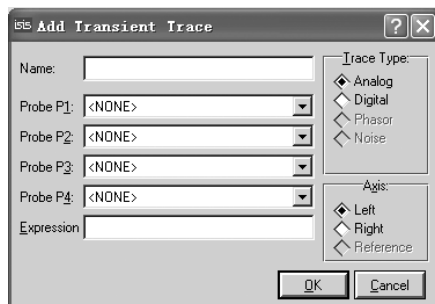


图 3-69 Add Trace 编辑对话框

在 Trace Type 中选择 Digital 型曲线，并在 Probe P1 中选择对应的数字探针，然后单击“OK”按钮，可将输入模式信号源添加到混合图表中。

其他探针的添加可以用菜单命令,也可采用拖动的方法,如图 3-70 所示。

(2) 设置仿真时间为 10s, 使用空格键开始仿真。电路的仿真结果如图 3-71 所示。

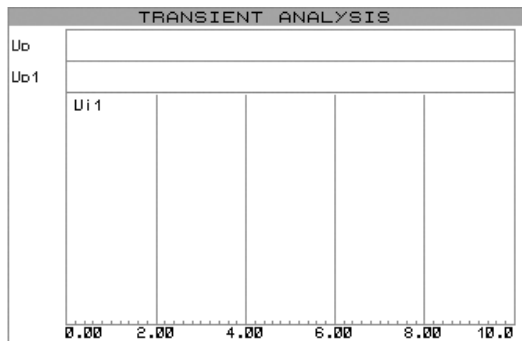


图 3-70 添加探针后的图表

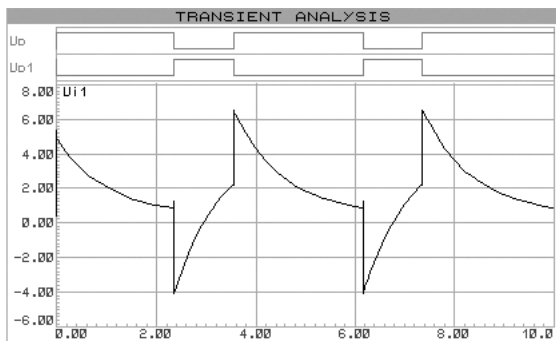


图 3-71 混合图表仿真结果图

3.2.4 基于频率分析图表的电路分析

频率分析的作用是分析电路在不同频率工作状态下的运行情况。但不像频谱分析仪,所有频率一起被考虑,而是每次只可分析一个频率。所以,频率特性分析相当于在输入端接一个可改变频率的测试信号,在输出端接一个交流电表测量不同频率所对应的输出,同时可得到输出信号的相位变化情况。频率特性分析还可以用来分析不同频率下的输入、输出阻抗。

此功能在非线性电路中使用是没有实际意义的。因为频率特性分析的前提是假设电路为线性的,就是说,如果在输入端加一标准的正弦波,在输出端也相应地得到一标准的正弦波。实际中完全线性的电路是不存在的,但是大多数我们认为是线性电路是在此分析允许范围内的。另外,由于系统是在线性情况下,且引入复数算法(矩阵算法)进行的运算,其分析速度要比瞬态分析快了许多。

频率分析图表用于绘制小信号电压增益或电流增益随频率变化的曲线,即绘制波特图,可描绘电路的幅频特性和相频特性。但它们都以指定的输入发生器为参考。在进行频率分析时,图表的 X 轴表示频率,两个纵轴可分别显示幅值和相位。

1. 放置频率分析图表

单击工具箱中的 Simulation Graph 图标,在对象选择器中选择 FREQUENCY 仿真图表。在编辑窗口期望放置图表的位置单击鼠标左键,并拖动鼠标,在期望的结束点单击鼠标左键,放置频率分析图表,如图 3-72 所示。

2. 设置频率分析图表

双击图表将弹出如图 3-73 所示的频率分析图表编辑对话框。对话框中包含如下设置内容:

☺ Graph title: 图表标题。

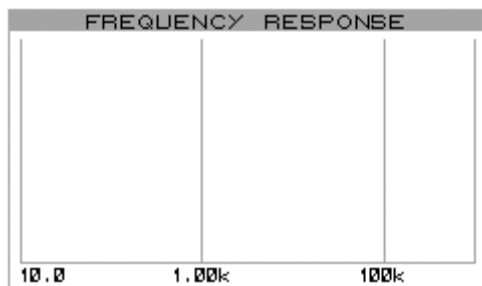


图 3-72 频率分析图表

- ☺ Reference: 参考发生器。
- ☺ Start frequency: 起始仿真频率。
- ☺ Stop frequency: 终止仿真频率。
- ☺ Interval: 间距取值方式。系统提供三种取值方式:
 - ☞ DECADES 十倍频程;
 - ☞ OCTAVESL 八倍频程;
 - ☞ INEAR 线性取值。
- ☺ No.Steps/Interval: 步幅数。

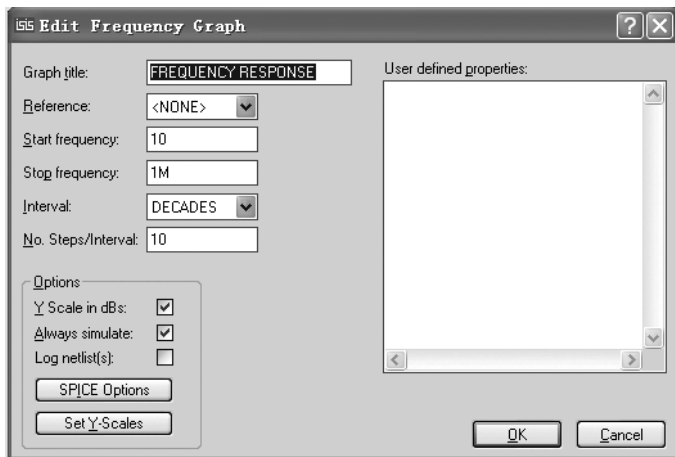


图 3-73 频率分析图表编辑对话框

在频率分析中,幅值和相位都需设一个参考值,我们通过设置参考发生器来实现这一点。图表中参考发生器的幅值为 1V,相位为 0dB。原理图中的任何一个发生器都可被设为参考发生器,图中的其他发生器在频率分析时被忽略。

在进行仿真时一定要选择一个参考发生器并把它拖到图中,否则仿真不能正确进行。编辑完成后,单击“OK”按钮完成设置。

3. 进行仿真

本例中使用图 3-74 所示的电路进行仿真。频率分析图表的设置如表 3-12 所示。

表 3-12 频率分析图表设置

标 题	参考发生器	起始频率	终止频率	间距取值方式	步长
FREQUENCY RESPONSE	INPUT	10Hz	1MHz	DECADES	10

执行 Graph→Simulate (快捷键: 空格键) 命令,开始仿真。此时,图表也随仿真的结果进行更新。电路仿真结果如图 3-75 所示。

如上所述,测量时,需放置两条测量线(平行于竖轴):在图表中左击鼠标,出现一条绿线;按下 Ctrl 键,在图表中左击鼠标,出现另一条红线。

移动测量线时也一样:左击鼠标移动绿线,按下 Ctrl 键左击鼠标则移动红线。

删除测量线:鼠标指向任一竖轴的标值(如左轴的-200m、400m 等)左击鼠标删除绿

线：鼠标指向任一竖轴的标值，按下 Ctrl 键，左击鼠标，删除红线。

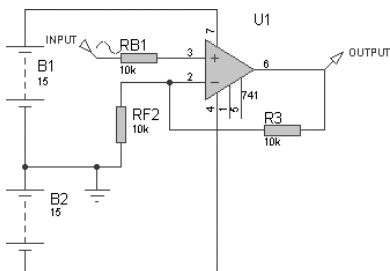


图 3-74 仿真电路

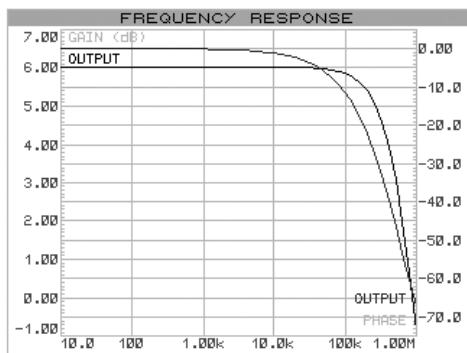


图 3-75 频率分析仿真结果

图表底部为状态栏，显示的数据都是绝对值。其中 DX 显示时间相对量；DY 显示幅值相对量。

3.2.5 基于转移特性分析图表的电路分析

转移特性分析是一种非线性分析，用于分析在给定激励信号情况下电路的时域响应。

1. 放置转移特性分析图表

单击工具箱中的 Simulation Graph 图标，在对象选择器中选择 TRANSFER 仿真图表。在编辑窗口期望放置图表的位置单击鼠标左键，并拖动鼠标，在期望的结束点单击鼠标左键，放置图表，如图 3-76 所示。

2. 设置转移特性分析图表

双击转移特性分析图表，将弹出如图 3-77 所示的转移特性分析图表编辑对话框。

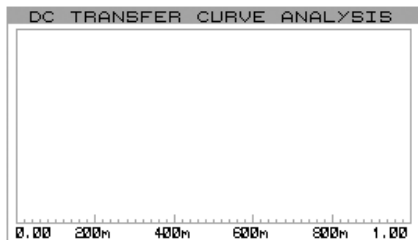


图 3-76 转移特性分析图表

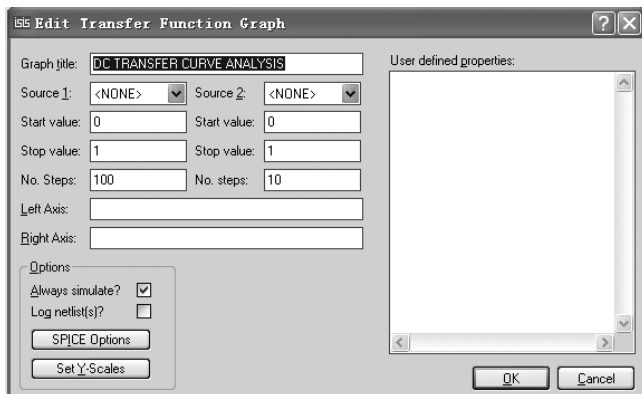


图 3-77 转移特性分析图表编辑对话框

☺ Graph title: 图表标题。

☺ Source 1: 横轴仿真源。

☺ Start value: 横轴仿真源仿真起始值。

Source 2: 激励源。

Start value: 激励源仿真起始值。

- ☺ Stop value: 横轴仿真源仿真终止值。 Stop value: 激励源仿真终止值。
 ☺ No.Steps: 步幅数。 No.steps: 步幅数。
 ☺ Left Axis: 左边坐标轴标签。 Right Axis: 右边坐标轴标签。
- 按照电路要求设置转移特性分析图表。编辑完成后, 单击“OK”按钮完成设置。

3. 进行仿真

本例中使用图 3-78 所示电路进行仿真 (其中, $V_{CE}=1V$)。

转移特性分析图表的设置如表 3-13 所示。

表 3-13 转移特性分析图表设置

图表标题	DC TRANSFER CURVE ANALYSIS		
源 1	V_{CE}	源 2	I_B
源 1 仿真起始值	0V	源 1 仿真起始值	100 μ V
源 1 仿真终止值	10V	源 1 仿真终止值	1mV
步长	100	步长	10

选择 Graph→Simulate (快捷键: 空格键) 命令, 开始仿真。此时, 图表也随仿真的结果进行更新。电路仿真结果如图 3-79 所示。

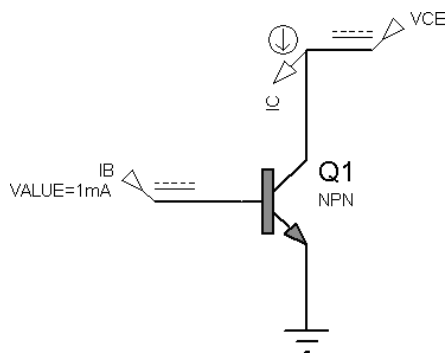


图 3-78 转移特性分析例图

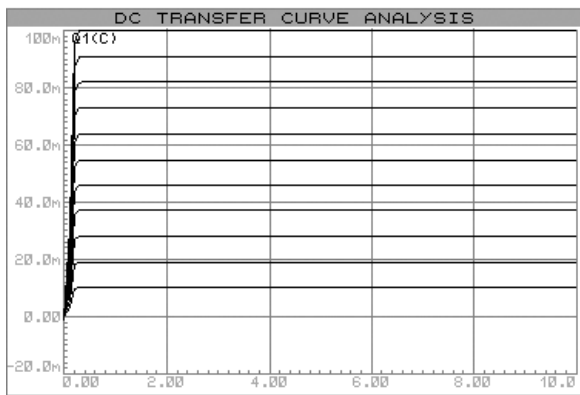


图 3-79 转移特性分析图表仿真结果图

3.2.6 基于噪声分析图表的电路分析

由于电阻或半导体元件会自然而然地产生噪声, 这对电路工作当然会产生相当程度的影响。系统提供噪声分析就是将噪声对输出信号所造成的影响予以数字化, 以供设计师评估电路性能。

在分析时, SPICE 模拟装置可以模拟电阻器及半导体元件产生的热噪声, 各元件在设置电压探针 (因为该分析不支持噪声电流, PROSPICE 不考虑电流探针) 处产生的噪声将在该点求和, 即为该点的总噪声。分析曲线的横坐标表示的是该分析所在的频率范围, 纵坐标表示的是噪声值 (分左、右 Y 轴, 左 Y 轴表示输出噪声值, 右 Y 轴表示输入噪声值。一般以 V/\sqrt{Hz} 为单位, 也可通过编辑图表对话框设置为 dB, 0dB 对应 $1V/\sqrt{Hz}$)。电路

工作点将按照一般处理方法计算，在计算工作点之外的各时间，除了参考输入信号外，各信号发生装置将不被分析系统考虑，所以，分析前不必移除各信号发生装置。PROSPICE 于分析过程中在计算所有电压探针噪声的同时考虑它们相互间的影响，所以无法知道单纯的某个探针的噪声分析结果。分析过程将对每个探针逐一处理，所以仿真时间大概与电压探针的数量成正比。

注意：

噪声分析是不考虑外部电、磁等对电路的影响的。

噪声分析图表可显示随频率变化时节点的等效输入、输出噪声电压，同时可产生单个元件的噪声电压清单。

1. 放置噪声分析图表

单击工具箱中的 Simulation Graph 图标，在对象选择器中选择 NOISE 仿真图表。在编辑窗口期望放置图表的位置单击鼠标左键，并拖动鼠标，在期望的结束点单击鼠标左键，放置噪声分析图表，如图 3-80 所示。

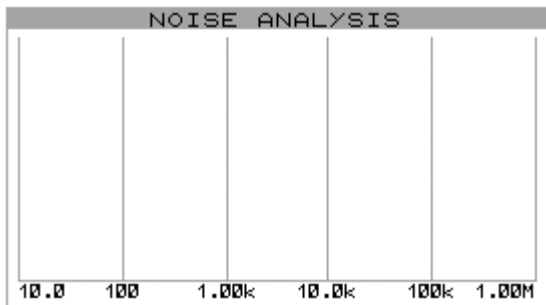


图 3-80 噪声分析图表

2. 设置噪声分析图表

双击图表将弹出如图 3-81 所示的噪声分析图表编辑对话框。

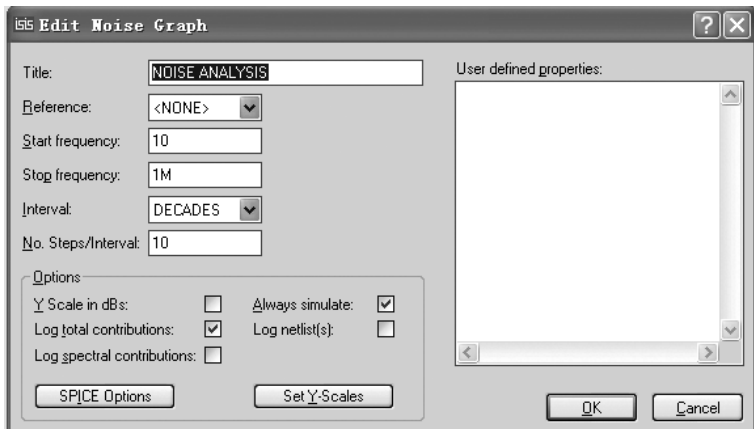


图 3-81 噪声分析图表编辑对话框

对话框中包含如下设置内容：

- ☺ Title: 图表标题。
- ☺ Reference: 参考发生器。
- ☺ Start frequency: 起始仿真频率。
- ☺ Stop frequency: 终止仿真频率。
- ☺ Interval: 间距取值方式。系统提供三种取值方式：
 - ☑ DECADES 十倍频程；
 - ☑ OCTAVESL 八倍频程；
 - ☑ INEAR 线性取值。
- ☺ No.Steps/Interval: 步幅数。

按照电路实际要求设置噪声分析图表。编辑完成后，单击“OK”按钮完成设置。

3. 进行仿真

本例中使用图 3-74 所示电路进行仿真。噪声分析图表的设置如表 3-14 所示。

表 3-14 噪声分析图表设置

标 题	参考发生器	起始频率	终止频率	间距取值方式	步长
NOISE ANALYSIS	INPUT	10Hz	1MHz	DECADES	10

选择 Graph→Simulate（快捷键：空格键）命令，开始仿真。此时，图表将随仿真的结果进行更新。电路仿真结果如图 3-82 所示。

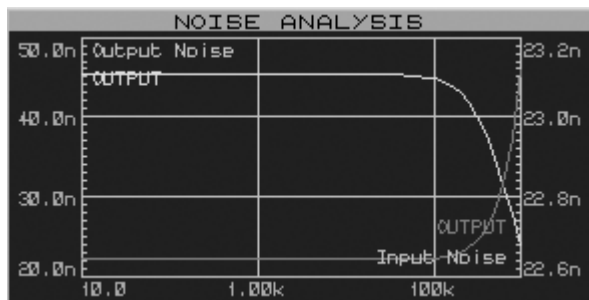


图 3-82 噪声分析仿真结果

3.2.7 基于失真分析图表的电路分析


失真是由电路传输函数中的非线性部分产生的，仅由线性元件组成的电路（如电阻、电感、线性可控源）不会产生任何失真。失真分析用于检测电路中的谐波失真和互调失真。

PROTEUS ISIS 的失真分析可仿真二极管、双极性晶体管、场效应管、面结型场效应晶体管（JFET）和金属氧化物半导体场效应晶体管（MOSFET），用于确定由测试电路所引起的电平失真程度。

对于单频率信号，PROTEUS ISIS 失真分析可确定电路中每一节点的二次谐波和三次谐波造成的失真；对于互调失真，即电路中有频率分别为 F_1 、 F_2 的交流信号源，则 PROTEUS ISIS 频率分析给出电路节点在 F_1+F_2 、 F_1-F_2 及 $2F_1-F_2$ 不同频率上的谐波失真。

失真分析对于研究瞬态分析中不易观察到的小失真比较有效。

1. 放置失真分析图表

单击工具箱中的 Simulation Graph  图标，在对象选择器中选择 DISTORTION 仿真图表。在编辑窗口期望放置图表的位置单击鼠标左键，并拖动鼠标，在期望的结束点单击鼠标左键，放置失真分析图表，如图 3-83 所示。

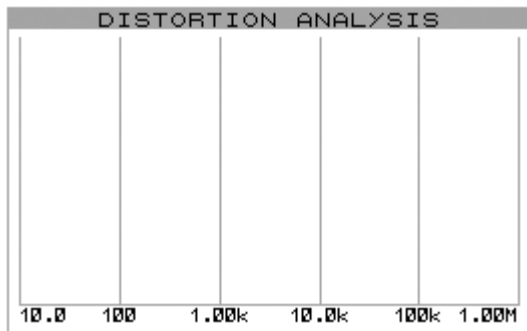


图 3-83 失真分析图表

2. 设置失真分析图表

双击图表将弹出如图 3-84 所示的失真分析图表编辑对话框。

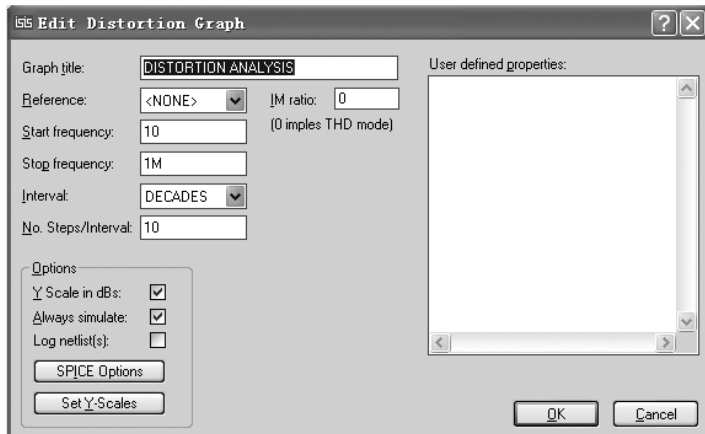


图 3-84 失真分析图表编辑对话框

对话框中包含如下设置内容：

- ☺ Graph title: 图表标题。
- ☺ Reference: 频率为 F_1 的发生器。
- ☺ IM ratio: F_2 与 F_1 的比率。
- ☺ Start frequency: F_1 起始仿真频率。
- ☺ Stop frequency: F_1 终止仿真频率。
- ☺ Interval: 间距取值方式。系统提供三种取值方式：
 - ☞ DECADES 十倍频程；

☞ OCTAVESL 八倍频程;

☞ INEAR 线性取值。

☺ No.Steps/Interval: 步幅数。

其中 IM ratio 在仿真电路的互调失真时用于设置 F_2 与 F_1 的比率; 此时设置的频率范围为 F_1 的频率范围, F_2 的频率范围为 F_1 的频率乘以 F_2 与 F_1 的比率; IM ratio 的值设置为 0~1 之间的数。当 IM ratio 设置为 0 时, 系统仿真电路的谐波失真。

按照电路实际要求设置失真分析图表。编辑完成后, 单击“OK”按钮完成设置。

3. 进行仿真

本例中使用图 3-74 所示电路进行仿真。失真分析图表的设置如表 3-15 所示。

表 3-15 失真分析图表设置

标 题	参考发生器	起始频率	终止频率	间距取值方式	步 长	IM ratio
DISTORTION ANALYSIS	INPUT	10Hz	1MHz	DECADES	10	0

选择 Graph→Simulate (快捷键: 空格键) 命令, 开始仿真。图表也随仿真的结果进行更新。电路仿真结果如图 3-85 所示。

失真分析将生成每个谐波的幅值和相位信息。

对于单频率谐波畸变, 在图表中将出现两条曲线, 分别表示信号的二次谐波和三次谐波。

对于互调失真, 将按照二次谐波与基波的比率输入两个频率信号。每一轨迹将有三条曲线分别显示 F_1+F_2 、 F_1-F_2 和 $2F_1-F_2$ 。

在定义二次谐波与基波的比率时须小心。因为, 如果 $F_2/F_1=0.5$, 则 F_1-F_2 值等于 F_2 , 即且 F_1-F_2 曲线将没有意义。在通常状况下, 这一比率值取无理数。如 $F_2/F_1=49/100$ 是一比较好的取值。

📖 注意:

$F_2/F_1 < 1$ 。

无论在何种失真分析下, 将指针放置到曲线图表上后, 在状态栏的右边都将会显示曲线的类型, 如图 3-86 所示。

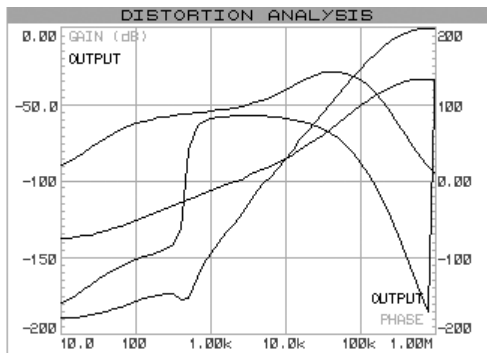


图 3-85 失真分析仿真结果

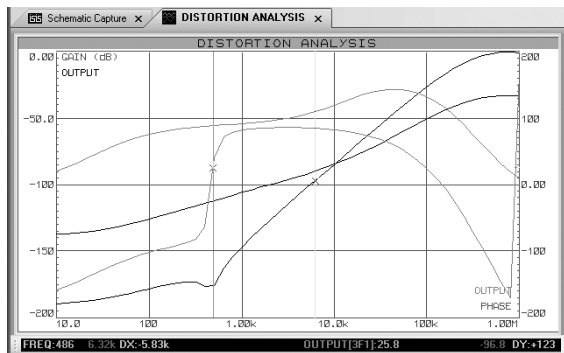


图 3-86 失真曲线的状态显示

3.2.8 基于傅里叶分析图表的电路分析

傅里叶分析方法用于分析一个时域信号的直流分量、基波分量和谐波分量。即把被测节点处的时域变化信号作离散傅里叶变换，求出它的频域变换规律，将被测节点的频谱显示在分析图窗口中。在进行傅里叶分析时，必须首先选择被分析的节点，一般将电路中的交流激励源的频率设为基频，若在电路中有几个交流电源，则可将基频设为电源频率的最小公因数。

PROTEUS ISIS 系统为模拟电路频域分析提供了傅里叶分析图表。系统首先对电路进行瞬态分析，然后对瞬态分析结果执行快速傅里叶分析（FFT）。为了优化 FFT 分析，在仿真图表中提供了多种窗函数。

由傅里叶分析计算系统失真度（ D ）的计算公式为： $D \approx \sqrt{\frac{V_{om2}^2 + V_{om3}^2}{V_{om1}^2}}$ ，其中 V_{om1}^2 是基波幅度，而 V_{om2}^2 、 V_{om3}^2 为二次谐波与三次谐波。

1. 放置傅里叶分析图表

单击工具箱中的 Simulation Graph 图标，在对象选择器中选择 FOURIER 仿真图表。在编辑窗口期望放置图表的位置单击鼠标左键，并拖动鼠标，在期望的结束点单击鼠标左键，放置傅里叶分析图表，如图 3-87 所示。

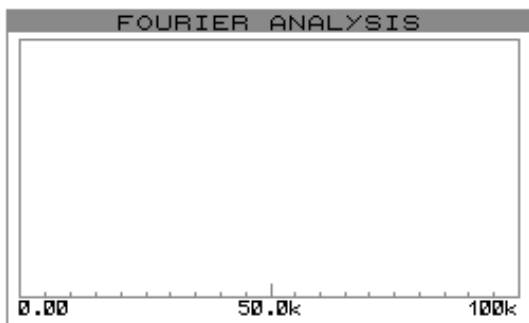


图 3-87 傅里叶分析图表

2. 设置傅里叶分析图表

双击图表将弹出如图 3-88 所示的傅里叶分析图表编辑对话框。

对话框中包含如下设置内容：

- ☺ Graph title: 图表标题。
- ☺ Start time: 仿真起始时间。
- ☺ Stop time: 仿真终止时间。
- ☺ Max Frequency: 最大频率。
- ☺ Resolution: 分辨率。
- ☺ Window: 窗函数。
- ☺ Left Axis Label: 左边坐标轴标签。

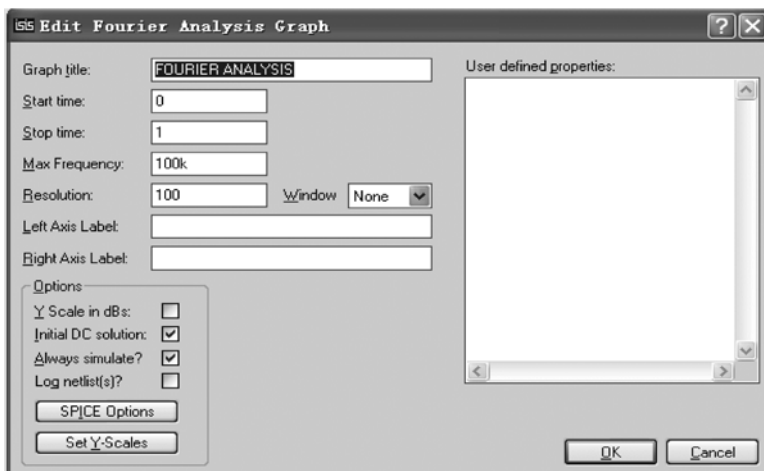


图 3-88 傅里叶分析图表编辑对话框

☺ Right Axis Label: 右边坐标轴标签。

按照电路实际要求设置傅里叶分析图表。编辑完成后，单击“OK”按钮完成设置。

3. 进行仿真

本例中使用图 3-89 所示电路进行仿真。

傅里叶分析图表的设置如表 3-16 所示。

表 3-16 傅里叶分析图表设置

图表标题	起始时间	终止时间	最大频率	分辨率	窗函数
FFT OF REAL SQUARE WAVE	0s	100ms	20kHz	125	None

选择 Graph→Simulate（快捷键：空格键）命令，开始仿真。此时，图表也随仿真的结果进行更新。电路仿真结果如图 3-90 所示。

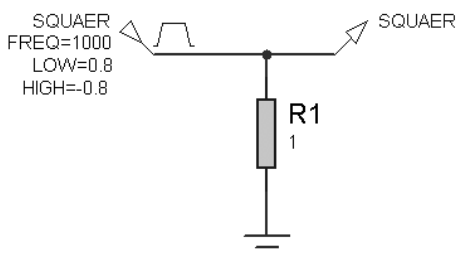


图 3-89 傅里叶分析例图

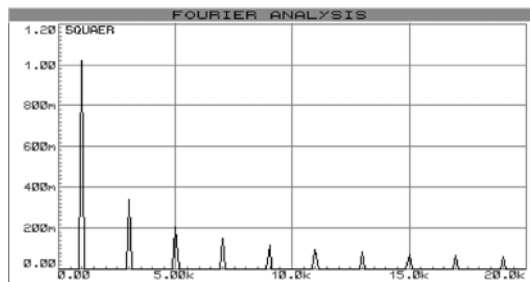



图 3-90 傅里叶分析曲线

3.2.9 基于音频分析图表的电路分析

音频分析用于用户从设计的电路中听电路的输出（要求系统具有声卡）。实现这一功能的主要元件为音频分析图表。这一分析图表与模拟分析图表在本质上是一样的，只是在仿真结束后，会生成一个时域的 WAV 文件窗口，并且可通过声卡输出声音。

1. 放置音频分析图表

单击工具箱中的 Simulation Graph 图标，在对象选择器中选择 AUDIO 仿真图表。在编辑窗口期望放置图表的位置单击鼠标左键，并拖动鼠标，在期望的结束点单击鼠标左键，放置音频分析图表，如图 3-91 所示。

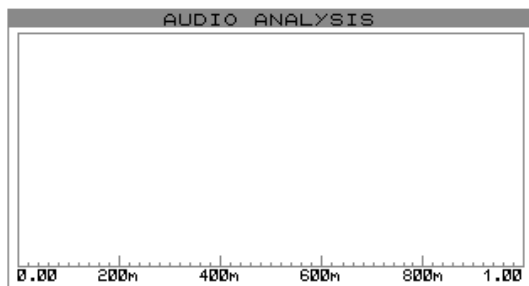


图 3-91 音频分析图表

2. 设置音频分析图表

在图表中放置测量探针，设置音频分析图表，如图 3-92 所示。

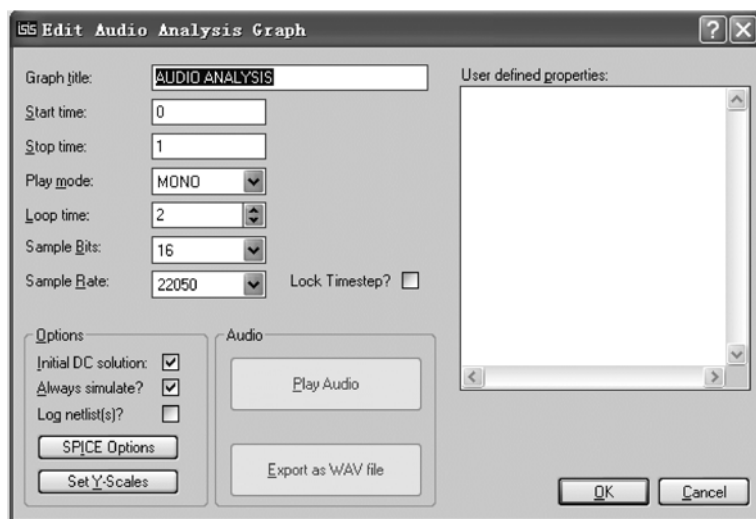


图 3-92 设置音频分析图表

对话框中包含如下设置内容：

- ☺ Graph title: 图表标题。
- ☺ Start time: 仿真起始时刻。
- ☺ Stop time: 仿真终止时刻。
- ☺ Play mode: 播放模式。系统提供了 MONO、INPUT、OUTPUT 和 STEREO 四种播放模式。
- ☺ Loop time: 循环时间。
- ☺ Sample Bits: 采样位。系统提供了 8 和 16 两种采样位。

☺ Sample Rate: 采样率。系统提供了 11 025、22 050、44 100 三种采样率。按电路实际要求编辑完成后,单击“OK”按钮完成设置。

3. 进行仿真

本例中使用图 3-93 所示电路进行仿真。

音频分析图表的设置如表 3-17 所示。

表 3-17 音频分析图表设置

图表标题	起始时间	终止时间	播放模式	循环时间	采样位	采样率
AUDIO ANALYSIS	0s	1ms	MONO	2	16	44 100

选择 Graph→Simulate (快捷键: 空格键) 命令,开始仿真。此时,图表也随仿真的结果进行更新。电路仿真结果如图 3-94 所示。

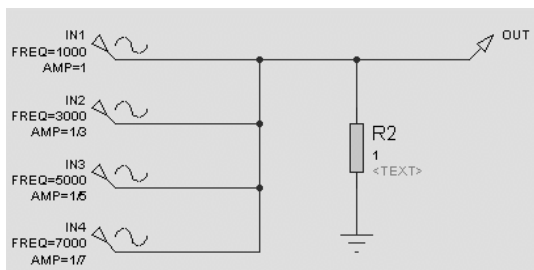


图 3-93 音频分析例图

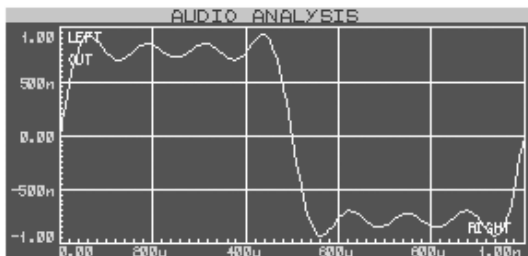


图 3-94 音频分析曲线

根据在最短的实际仿真内生成合适长度的波形,选择启动、停止时间和循环时间。通过 1ms 的分析和 1 000 次的循环创建的 1s 音频将明显比 1s 的电路分析快得多。

为信号选择一个采样分辨率和采样率,以使信号不失真。在不需要创建大的文件,或缺少磁盘空间的情况下,使用 16 位的分辨率。大多数 PC 的扬声器系统选择 43.1kHz 的分辨率。

按空格键,调用 PROSPICE。按“Ctrl+ Space”键,重播音频文件,而不进行仿真。

3.2.10 基于交互分析图表的电路分析

交互式分析结合了交互式仿真与图表仿真的特点,仿真过程中,系统建立交互式模型,但分析结果却是用一个瞬态分析图记录 and 显示的。交互式分析特别适用于观察电路中的某一单独操作对电路产生的影响(如变阻器阻值变化对电路的影响情况),相当于将一个示波器和一个逻辑分析仪结合在一个装置上。

分析过程中,系统按照混合模型瞬态分析的方法进行运算,但仿真是在交互式模型下运行的。因此,像开关、键盘等各种激励的操作将对结果产生影响。同时,仿真速度也取决于交互式仿真中设置的时间步长。

注意:

在分析过程中,系统将获得大量数据,处理器每秒将会产生数百万事件,产生的各种事情将占用许多兆内存,这就很容易使系统崩溃,所以不宜进行长时间仿真。也就是

说，短时间仿真不能实现目的时，应用逻辑分析仪。另外，和普通交互式仿真不同的是，许多成分电路不被该分析支持。

通常情况，可以借助交互式仿真中的虚拟仪器实现观察电路中的某一单独操作对电路产生的影响，但有时需要将结果用图表的方式显示出来以便更详细地分析，这就需要用交互式分析来实现。

1. 放置交互分析图表

单击工具箱中的 Simulation Graph 图标，在对象选择器中选择 INTERACTIVE 仿真图表。在编辑窗口期望放置图表的位置单击鼠标左键，并拖动鼠标，在期望的结束点单击鼠标左键，放置交互分析图表，如图 3-95 所示。

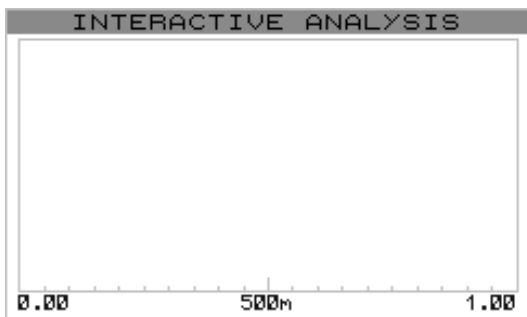


图 3-95 交互分析图表

2. 设置交互分析图表

双击交互分析图表，将弹出如图 3-96 所示的交互分析图表编辑对话框。

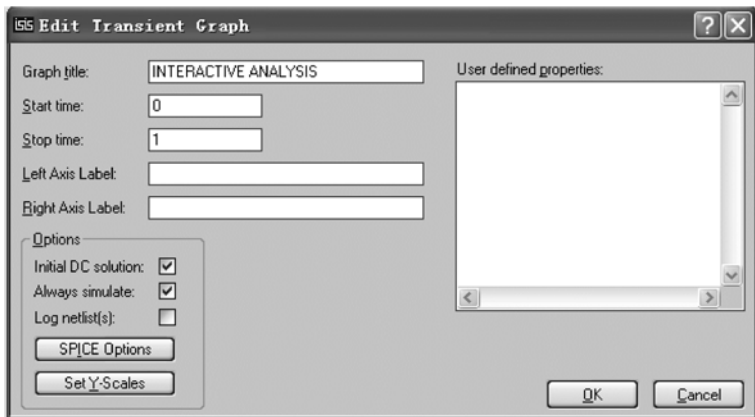


图 3-96 交互分析图表编辑对话框

对话框中包含如下设置内容：

- ☺ Graph title: 图表标题。
- ☺ Start time: 仿真起始时刻。
- ☺ Stop time: 仿真终止时刻。

☺ Left Axis Label: 左边坐标轴标签。

☺ Right Axis Label: 右边坐标轴标签。

设置完成后, 单击“OK”按钮, 结束设置。

3. 进行仿真

本例中使用图 3-68 所示电路进行仿真。

选择 Graph→Simulate (快捷键: 空格键) 命令, 开始仿真。此时, 图表也随仿真的结果进行更新。电路仿真结果如图 3-97 所示。

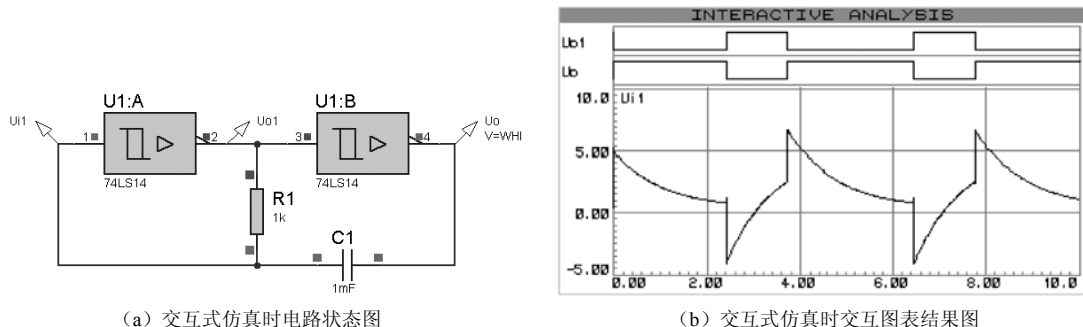


图 3-97 电路仿真结果

3.2.11 基于一致性分析图表的电路分析

一致性分析用于比较两组数字仿真结果。这一分析图表可以快速测试改进后的设计是否带来不期望的副作用。一致性分析作为测试策略的一部分, 通常应用于嵌入式系统的分析中。

一致性分析可以在图表中保存两组结果, 并分别称其为测试结果与参考结果。

电路是否具有 consistency, 由初始迹线 (first trace) 以不同的颜色显示以使其与其他曲线区分开来。

1. 放置一致性分析图表

单击工具箱中的 Simulation Graph 图标, 在对象选择器中将出现各种仿真分析所需的图表, 选择 CONFORMANCE 仿真图表。

在编辑窗口期望放置图表的位置单击鼠标左键, 并拖动鼠标, 在期望的结束点单击鼠标左键, 放置图表, 如图 3-98 所示。

双击一致性分析图表, 将弹出如图 3-99 所示的一致性分析图表编辑对话框。

对话框中包含如下设置内容:

☺ Graph title: 图表标题。

☺ Start time: 仿真起始时刻。

☺ Stop time: 仿真终止时刻。

☺ Left Axis Label: 左边坐标轴标签。

☺ Right Axis Label: 右边坐标轴标签。

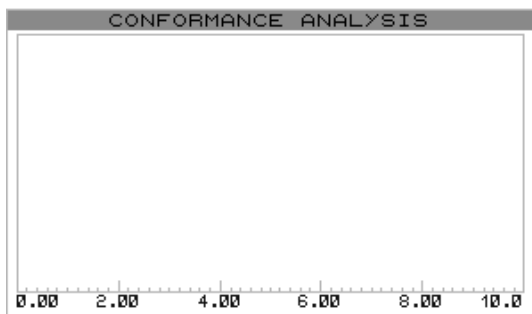


图 3-98 一致性分析图表

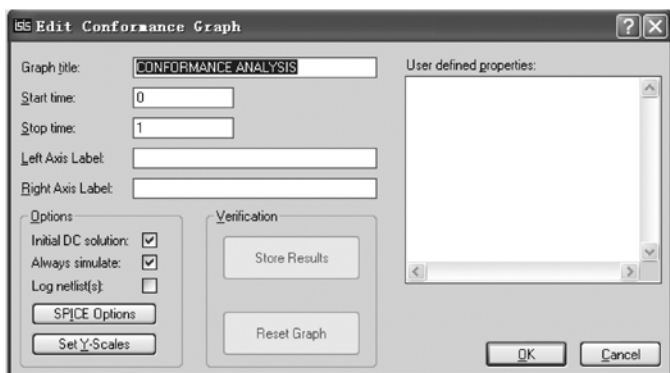


图 3-99 一致性分析图表编辑对话框

根据电路实际需要设置一致性分析图表，编辑完成后，单击“OK”按钮完成设置。

2. 建立电路

本例中使用图 3-100 所示电路进行一致性分析。

这一电路由 U1、R1 和 C1 构成了典型的 555 单稳态触发器。这一电路由数字脉冲发生器在 1ms 处触发，其中 TP 的设置 Start Time (Secs): 1ms; Pulse Width: 205 μ s。同时结果输出波形显示在模拟分析图表中。

电路的模拟仿真结果如图 3-101 所示。

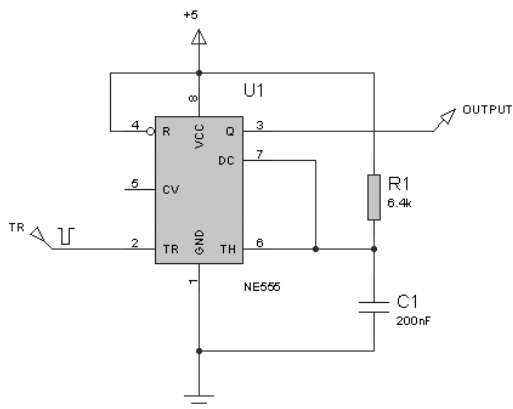


图 3-100 一致性分析例图

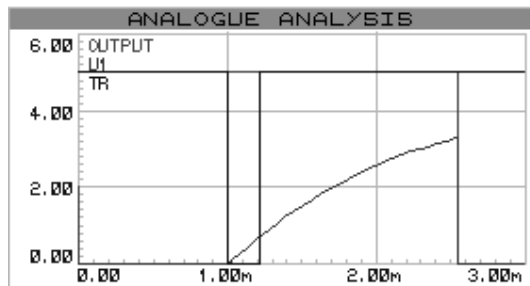


图 3-101 一致性分析例图模拟仿真结果图

电路仿真结果应具有以下特性:

- ☺ 在触发发生之前, 输出为低电平。
- ☺ 触发端变为低电平后, 输出马上变为高电平。
- ☺ 当触发端变回高电平时, 输出仍保持为高电平。
- ☺ 输出保持高电平约 1.5ms, 之后输出变为低电平。

3. control trace 的获取

为了完成一致性分析, 需在触发信号的每一边沿及输出信号的每一边沿采样输出数据。本例中采用图 3-102 所示电路获取 control trace。

其中, 数字单脉冲信号源指标如表 3-18 所示。

表 3-18 数字单脉冲信号源指标

Generator Name	Pulse Polarity	Start Time(Secs)	Pulse Width(Secs)
U2 (D0)	Positive (Low-High-Low) Pulse	995m	10μ
U2 (D1)	Positive (Low-High-Low) Pulse	1.2m	10μ
U2 (D2)	Positive (Low-High-Low) Pulse	2.4m	200μ

用户通过选择最后两个采样点的间距, 可以设置延迟时间的容限。

用户可使用数字时钟发生器或模式发生器产生控制信号; 采用单时钟信号发生器将使电路每隔一定间隔进行查证, 因此常采用这一方法。

放置一致性分析图表, 确保 control trace 在图表的顶端。

将 CONTROL、TR、OUTPUT 指针放置到一致性分析图表中, 并进行编辑。一致性分析图表编辑对话框如图 3-99 所示。电路仿真结果图见图 3-103。

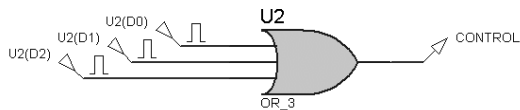


图 3-102 获取 control trace 电路

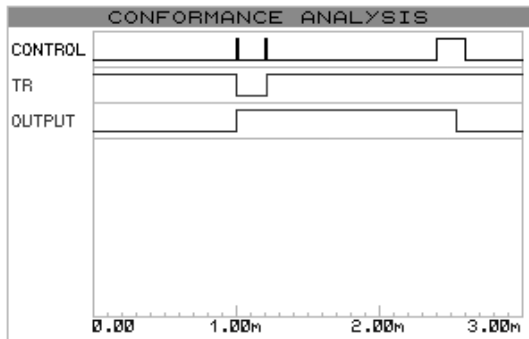


图 3-103 电路仿真结果图

验证结果是否是期望的结果相同, control trace 在期望的时间点处的跳变是否与期望的结果相同。

打开一致性分析图表编辑对话框, 并单击“Store Results”按钮。这一操作将使得当前显示的结果成为参考结果, 且图线将以暗色显示, 如图 3-104 所示。

保存设计。现假设元件不足, R_1 和 C_1 值须改变。例如 $R_1 = 15k\Omega$ 和 $C_1 = 100nF$, 此时对设计进行一致性分析。试验的结果如图 3-105 所示。

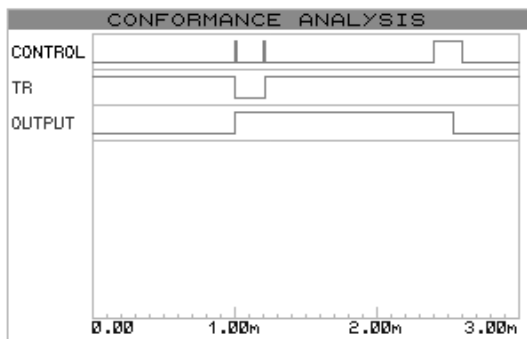


图 3-104 将获取的 control trace 保存

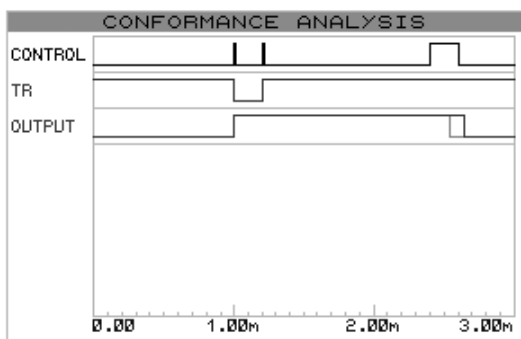



图 3-105 更改变量值后系统的仿真结果图

3.2.12 基于直流扫描分析图表的电路分析

直流扫描分析可以观察电路元件参数值在使用者定义范围内发生变化时对电路工作状态（电压或电流）的影响（如观察电阻值、晶体管放大倍数、电路工作温度等参数变化对电路工作状态的影响），也可以通过扫描激励元件参数值测量元件直流传输特性。

PROTEUS ISIS 系统为模拟电路分析提供了直流扫描图表，使用该图表，可以显示随扫描变化的定态电压或电流值。

1. 放置直流扫描分析图表

单击工具箱中的 Simulation Graph  图标，在对象选择器中将出现各种仿真分析所需的图表，选择 DC SWEEP 仿真图表。

在编辑窗口期望放置图表的位置单击鼠标左键，并拖动鼠标，在期望的结束点单击鼠标左键，放置图表，如图 3-106 所示。

2. 设置直流扫描分析图表

双击直流扫描分析图表，将弹出如图 3-107 所示的直流扫描分析图表编辑对话框。

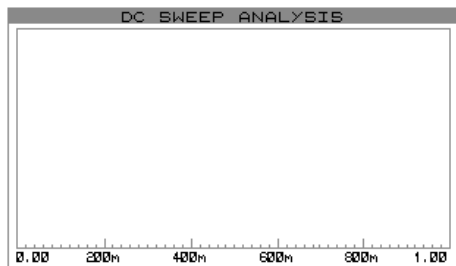


图 3-106 直流扫描分析图表

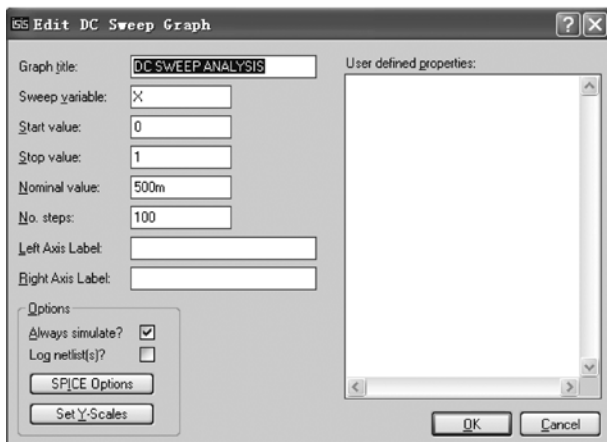


图 3-107 直流扫描分析图表编辑对话框

对话框中包含如下设置内容：

- ☺ Graph title: 图表标题。
- ☺ Sweep variable: 扫描变量。
- ☺ Start value: 扫描变量起始值。
- ☺ Stop value: 扫描变量终止值。
- ☺ Nominal value: 标称值。
- ☺ No.steps: 步幅数。
- ☺ Left Axis Label: 左边坐标轴标签。
- ☺ Right Axis Label: 右边坐标轴标签。

根据电路实际需要设置直流扫描分析图表，编辑完成后，单击“OK”按钮完成设置。

3. 进行仿真

本例中使用图 3-108 所示电路进行仿真。

直流扫描分析图表的设置如表 3-19 所示。

表 3-19 直流扫描分析图表设置

Graph title	Sweep variable	Start value	Stop value	Nominal value	No.steps
DIODE CHARACTERISTIC	V	-800mV	800mV	0	50

选择 Graph→Simulate（快捷键：空格键）命令，开始仿真。此时，图表也随仿真的结果进行更新。电路仿真结果如图 3-109 所示。

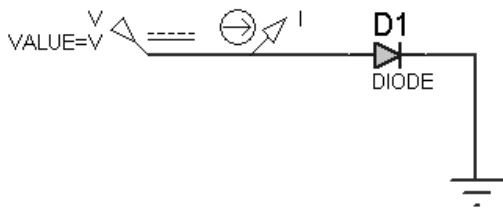


图 3-108 直流扫描分析例图

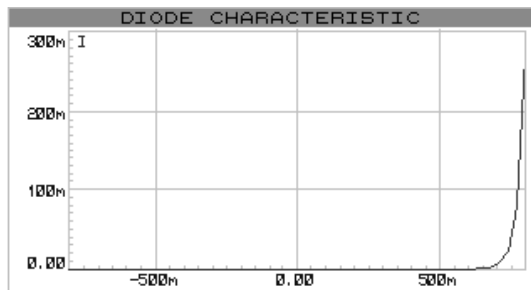


图 3-109 直流扫描分析仿真结果图


3.2.13 基于交流扫描分析图表的电路分析

交流扫描分析图表可以建立一组反映元件在参数值发生线性变化时的频率特性曲线，主要用来观测相关元件参数值发生变化时对电路频率特性的影响。

交流扫描分析时，系统内部完全按照普通的频率特性分析计算有关值，不同的是由于元件参数不固定而增加了运算次数，每次相应地计算一个元件参数值对应的结果。

与频率特性分析相同，左、右 Y 轴分别表示幅度（dB）、相位值。

1. 放置交流扫描分析图表

单击工具箱中的 Simulation Graph 图标, 在对象选择器中选择 AC SWEEP 仿真图表。

在编辑窗口期望放置图表的位置单击鼠标左键，并拖动鼠标，在期望的结束点单击鼠标左键，放置图表，如图 3-110 所示。

2. 设置交流扫描分析图表

双击交流扫描分析图表，将弹出如图 3-111 所示的交流扫描分析图表编辑对话框。

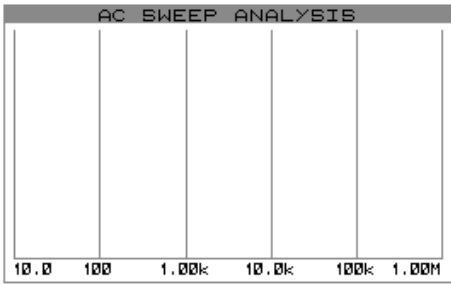


图 3-110 交流扫描分析图表

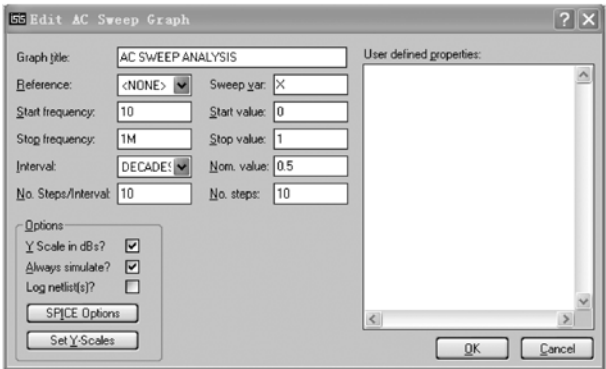


图 3-111 交流扫描分析图表编辑对话框

对话框中包含如下设置内容，见表 3-20。

表 3-20 设置内容

Graph title: 图表标题		
Reference: 参考信号源		Sweep variable: 扫描变量
Start frequency: 参考信号源仿真起始频率		Start frequency: 扫描变量仿真起始频率
Stop frequency: 参考信号源仿真终止频率		Stop frequency: 扫描变量仿真终止频率
Interval: 间距取值方式	DECADES 十倍频程	Nom.value: 标称值
	OCTAVESL 八倍频程	
	INEAR 线性取值	
No.Steps/Interval: 步幅数		No.steps: 步幅数

按照电路要求设置交流扫描分析图表。编辑完成后，单击“OK”按钮完成设置。

3. 进行仿真

本例中使用图 3-112 所示电路进行仿真。

交流扫描分析图表的设置如表 3-21 所示。

表 3-21 交流扫描分析图表设置

Graph title	AC SWEEP ANALYSIS		
Reference	R1(1)	Sweep variable	X
Start frequency	10Hz	Start value	1
Stop frequency	1MHz	Stop value	10
Interval	DECADES	Nom.value	5.5
No.Steps/Interval	100	No.steps	6

选择 Graph→Simulate（快捷键：空格键）命令，开始仿真。此时，图表也随仿真的结果进行更新。电路仿真结果如图 3-113 所示。

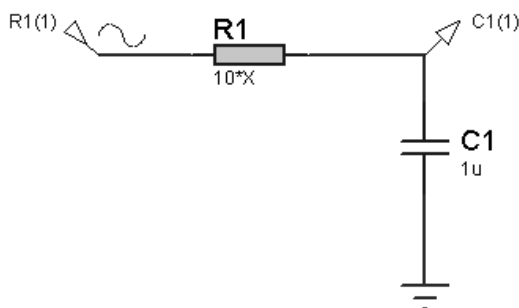


图 3-112 交流扫描分析例图

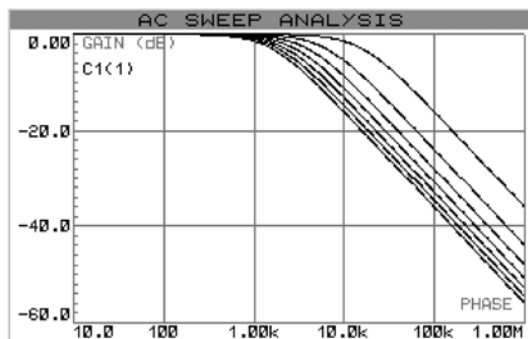


图 3-113 交流扫描分析仿真结果图

3.3 虚拟仪器


PROTEUS ISIS 为用户提供了多种虚拟仪器，单击工具箱中的按钮，在对象拾取器中列出所有的虚拟仪器名称，如图 3-114 所示。虚拟仪器及其含义见表 3-22。

表 3-22 虚拟仪器及其含义

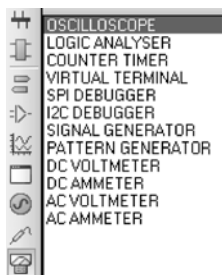



图 3-114 虚拟仪器列表

名 称	含 义
OSCILLOSCOPE	示波器
LOGIC ANALYSER	逻辑分析仪
COUNTER TIMER	计数器/定时器
VIRTUAL TERMINAL	虚拟终端
SPI DEBUGGER	SPI 调试器
I2C DEBUGGER	I ² C 调试器
SIGNAL GENERATOR	信号发生器
PATTERN GENERATOR	模式发生器
DC VOLTMETER	直流电压表
DC AMMETER	直流电流表
AC VOLTMETER	交流电压表
AC AMMETER	交流电流表

3.3.1 虚拟示波器（Oscilloscope）

（1）在 PROTEUS ISIS 环境中单击工具箱中的 Virtual Instrument Mode 按钮图标，出现如图 3-114 所示的虚拟仪器列表。

（2）单击“OSCILLOSCOPE”，则在预览窗口出现示波器的符号。

（3）在编辑窗口单击鼠标左键，出现示波器的拖动图像，拖动鼠标指针到合适位置，再次单击左键，示波器被放置到原理图编辑区中。虚拟示波器的原理图符号如图 3-115 所示。

(4) 示波器的四个接线端 A、B、C、D 应分别接四路输入信号，信号的另一端应接地。该虚拟示波器能同时观看四路信号的波形。

(5) 按图 3-116 所示连线。把 1kHz、1V 的正弦激励信号加到示波器 A 通道。

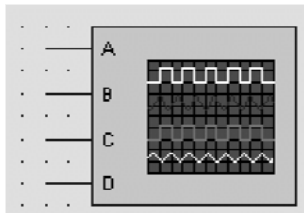


图 3-115 虚拟示波器的原理图符号

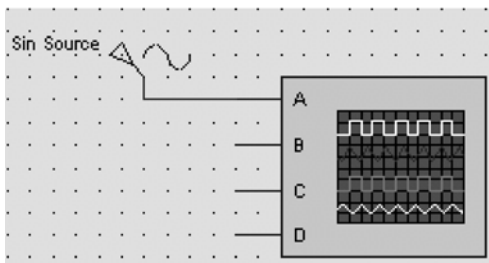



图 3-116 正弦信号与示波器的接法

(6) 单击仿真运行按钮  开始仿真，出现如图 3-117 所示的示波器运行界面。可以看到，左面的图形显示区有四条不同颜色的水平扫描线，其中 A 通道由于接了正弦信号，已经显示出正弦波形。

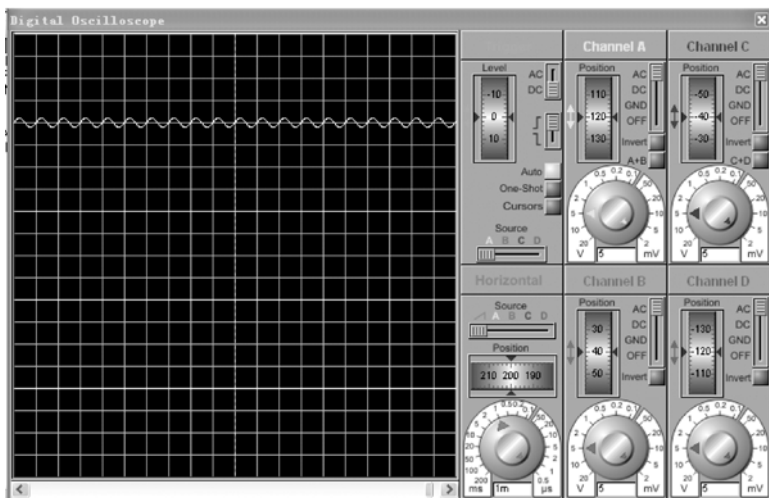


图 3-117 仿真运行后的示波器界面

(7) 示波器的操作区分以下几部分：

☺ Channel A: A 通道。

☺ Channel B: B 通道。

☺ Channel C: C 通道。

☺ Channel D: D 通道。



Position: 示波器显示垂直机械位置调节旋钮，用于调节所选通道波形的垂直位置。



选择开关: 选择通道显示波形类型。



旋钮: 用于调节垂直刻度系数。旋转图中的箭头可设置调节系统；另外，在文

本框中输入数据，按回车键也可设置调节系数。

☺ Trigger: 示波器触发信号设置，用于设置示波器触发信号的触发方式。



Level: 触发电平，用于调节电平。



选择开关: 触发电平类型。



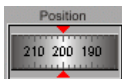
触发方式: 触发电平的触发方式。

☞ Auto: 自动设置触发方式。

☞ One-shot: 单击触发。

☞ Cursors: 选择指针模式。

☺ Horizontal: 示波器显示水平机械位置调节窗口。



滑动按钮: 用于调节波形的触发点位置。



旋钮: 用于调节水平比例尺因子。

【通道区】每个通道区的操作功能都一样。主要有两个旋钮，“Position”用来调整波形的垂直位移；下面的旋钮用来调整波形的 Y 轴增益，白色区域的刻度表示图形区每格对应的电压值。内旋钮是微调，粗旋钮是粗调。在图形区读波形的电压时，会把内旋钮顺时针调到最右端。

【触发区】其中“Level”用来调节水平坐标，水平坐标只在调节时才显示。“Auto”按钮一般为红色选中状态。“Cursors”光标按钮选中后，可以在图标区标注横坐标和纵坐标，从而度量波形的电压和周期，如图 3-118 所示。单击鼠标右键可以出现快捷菜单，选择清除所有的标注坐标、打印及颜色设置。

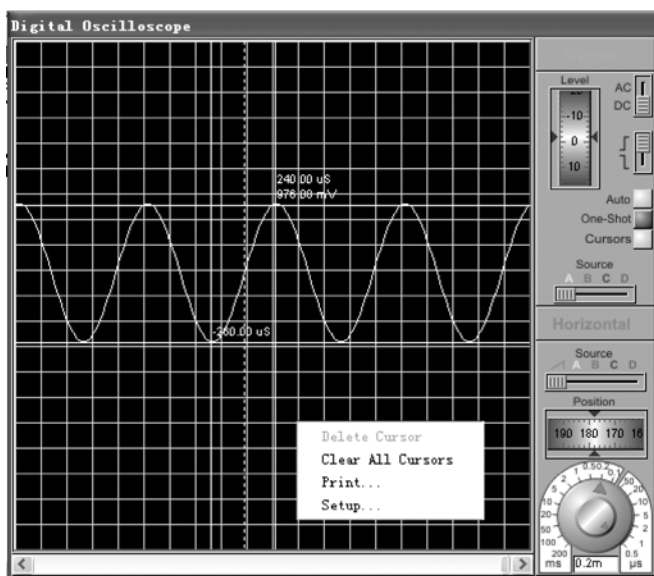


图 3-118 触发区“Cursors”按钮的使用

【水平区】“Position”用来调整波形的左右位移，下面的旋钮调整扫描频率。当读周期时，应把内环的微调旋钮顺时针转到底。

3.3.2 逻辑分析仪（Logic Analyser）

逻辑分析仪是通过将连续记录的输入信号存入到大的捕捉缓冲器进行工作的。这是一个采样过程，具有可调的分辨率，用于定义可以记录的最短脉冲。在触发期间，驱动数据捕捉处理暂停，并检测输入数据。触发前后的数据都可以显示。因其具有非常大的捕捉缓冲器（可存放 10 000 个采样数据），因此支持放大/缩小显示和全局显示。同时，用户还可移动测量标记，对脉冲宽度进行精确定时测量。

逻辑分析仪的原理图符号如图 3-119 所示。其中 A0~A15 为 16 路数字信号输入，B0~B3 为总线输入，每条总线支持 16 位数据，主要用于接单片机的动态输出信号。运行后，可以显示 A0~A15、B0~B3 的数据输入波形。

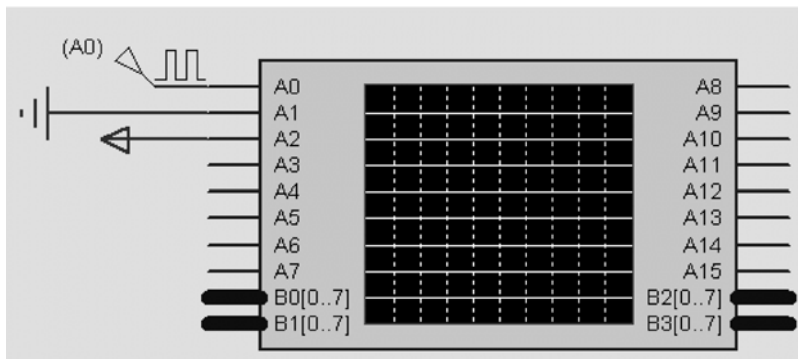


图 3-119 逻辑分析仪的原理图符号

逻辑分析仪的使用方法如下：

(1) 把逻辑分析仪放置到原理图编辑区，在 A0 输入端上接 10Hz 的方波信号，A1 接低电平，A2 接高电平。

(2) 单击仿真运行按钮，出现仿真操作界面，如图 3-120 所示。

(3) 先调整一个分辨率，类似于示波器的扫描频率，在图 3-121 中调捕捉分辨率“Capture Resolution”，单击光标按钮“Cursors”使其不再显示。单击捕捉按钮“Capture”，开始显示波形，该按钮变红，再变绿，稍后显示如图 3-121 所示的波形。

(4) 调整水平显示范围按钮“Display Scale”，或在图形区滚动鼠标滚轮，可调节波形，使其左右移动。

(5) 如果希望的波形没有出现，可以再次调整分辨率，然后单击捕捉按钮，就能重新生成波形。

(6) “Cursors”光标按钮按下后，在图形区单击，可标记横坐标的数值，即可以测出波形的周期、脉宽等。

从图 3-121 观察到，A0 通道显示方波，A1 通道显示低电平，A2 通道显示高电平，这两线紧挨着。其他没有接线的输入 A3~A15 一律显示低电平，B0~B3 由于不是单线而是总线，所以由两条高低电平来显示，如有输入，波形应为我们平时分析存储器读写时序时见到的数据或地址的波形。

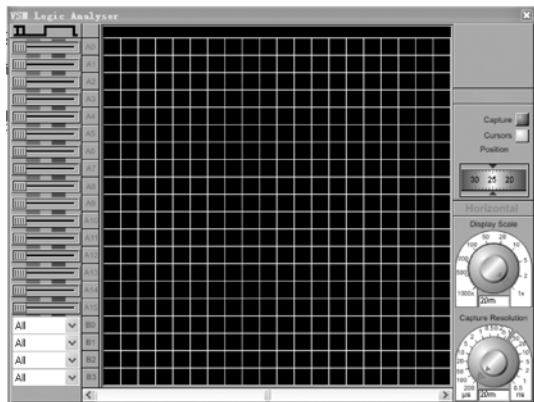


图 3-120 逻辑分析仪的仿真操作界面

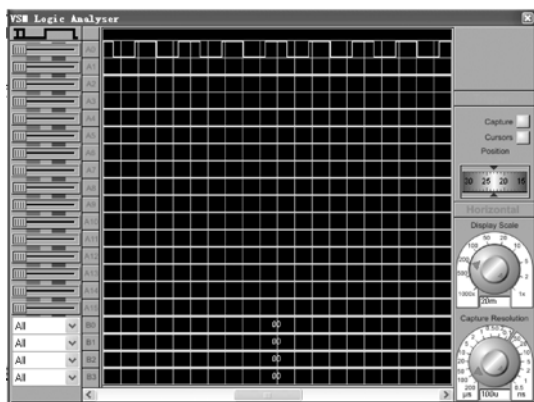


图 3-121 逻辑分析仪的仿真结果

3.3.3 计数器/定时器 (Counter Timer)

PROTEUS VSM 提供的计数器/定时器是一个通用的数字仪器，可用于测量时间间隔、信号频率和脉冲数。

计数器/定时器支持以下操作模式：

- ☺ 计时器方式（显示秒），分辨率为 $1\mu\text{s}$ 。
- ☺ 计时器方式（显示小时、分、秒），分辨率为 1ms 。
- ☺ 频率计方式，分辨率为 1Hz 。
- ☺ 计数器方式，最大计数值为 99 999 999。
- ☺ 计时值、频率数或计数值既在虚拟仪器界面显示，也在定时计数器的弹出式窗口显示。在仿真期间，选择菜单命令 Debug→VSM Counter Timer，即可出现弹出式窗口，如图 3-122 所示。

在这一弹出式窗口中，手动选择：

- ☺ RESET POLARITY: 复位电平极性；
- ☺ GATE POLARITY: 门信号极性；
- ☺ MANUAL RESET: 手动复位；
- ☺ MODE: 工作模式。

1. 使用计数器/定时器测量时间间隔

- (1) 单击工具箱中的 Virtual Instrument 按钮，则在对象选择窗口列出所包含的项目。
- (2) 从对象选择器中选择“COUNTER TIMER”，则在浏览窗口显示出虚拟计数器/定时器的图标。

- (3) 在编辑窗口单击鼠标左键，添加虚拟计数器/定时器，如图 3-123 所示。

- ☺ CE: 时钟使能引脚。当需要使能信号时，可将使能控制信号连接到这一引脚。如果不需要时钟使能，可将这一引脚悬空。
- ☺ RST: 复位引脚。这一引脚可使计时器复位、归零。如果不需要复位功能，也可将这一引脚悬空。
- ☺ CLK: 时钟引脚。



图 3-122 虚拟计数器/定时器弹出式窗口



图 3-123 虚拟时间间隔计数测量器

(4) 将鼠标放置在 Counter Timer 之上, 并使用快捷键 “Ctrl+E”, 打开编辑对话框进行设置, 如图 3-124 所示。

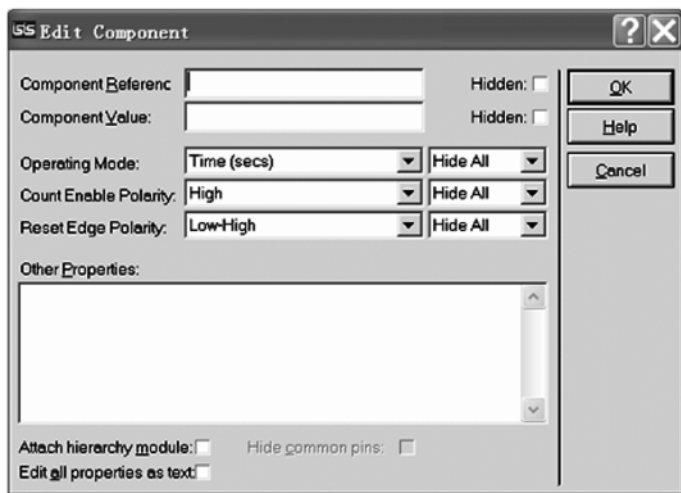


图 3-124 虚拟计数器/定时器编辑对话框

☺ Operating Mode: 工作模式选项。

☞ Default: 默认方式, 系统设置为计数方式。

☞ Time (secs): 定时方式, 相当于一个秒表, 最多计 100s, 精确到 1 μ s。CLK 端无须外加输入信号, 内部自动计时。由 CE 和 RST 端来控制暂停或重新从零开始计时。

☞ Time (hms): 定时方式, 相当于一个具有小时、分、秒的时钟, 最多计 10h, 精确到 1ms。CLK 端无须外加输入信号, 内部自动计时。由 CE 和 RST 端来控制暂停或重新从零开始计时。

☞ Frequency: 测频方式, 在 CE 有效和 RST 没有复位的情况下, 能稳定显示 CLK 外加的数字波频率。

☞ Count: 计数方式, 能够计外加时钟信号 CLK 的周期数, 最多计满八位, 即 99 999 999。

☺ Count Enable Polarity: 设置计数使能极性。

☺ Reset Edge Polarity: 复位信号边沿极性。

(5) 根据电路要求, 选择需要的计时模式 (secs 或 hms), 以及 CE 和 RST 功能的逻辑极性。

(6) 退出编辑窗口，运行仿真。

- ☺ 复位引脚 (RST pin) 为边沿触发方式，而不是电平触发方式。如果想要使定时计数器保持为零，可同时使用 CE 和 RST 引脚。
- ☺ 定时计数器的弹出式窗口提供了 manual reset (手动复位) 按钮。这一按钮可在仿真的任何时间复位计数器。这一功能在嵌入式系统中是非常有用的。使用这一功能，可以仿真程序的特定部分。

2. 使用计数器/定时器测量数字信号的频率

(1) 单击工具箱中的 Virtual Instrument 按钮，则在对象选择窗口列出所包含的项目。

(2) 从对象选择器中选择“COUNTER TIMER”，则在浏览窗口显示出虚拟计数器/定时器的图标。

(3) 在编辑窗口单击鼠标左键，添加虚拟计数器/定时器。

(4) 将待测信号连接到 CLK pin (时钟引脚)。在测量频率模式下，CE 和 RST 引脚无效。

(5) 将鼠标放置在 Counter Timer 之上，并使用快捷键“Ctrl+E”，打开编辑对话框，选择频率计方式。

(6) 退出编辑对话框，运行仿真。

- ☺ 频率计的工作原理为：在仿真期间计数每秒钟信号上升沿的数量，因此要求输入信号稳定，并且在完整的 1s 内有效。同时，如果仿真不是在实时速率下进行的（例如 CPU 超负荷运行），则频率计将在相对较长的时间内实时输出频率值。
- ☺ 计数器/定时器为纯数字器件。对于低电平模拟信号的频率测量，需要将待测信号通过 ADC 器件及其他逻辑开关，然后送入到计数器/定时器 CLK 引脚。同时，由于模拟仿真的速率只有数字仿真的 1/1 000，因而计数器/定时器不适合测量频率高于 10kHz 的模拟振荡电路的频率。在这种状况下，用户可以使用虚拟示波器（或图表）来测量信号周期。

3. 使用计数器/定时器计数数字脉冲

(1) 单击工具箱中的 Virtual Instrument 按钮，则在对象选择窗口列出所包含的项目。

(2) 从对象选择器中选择“COUNTER TIMER”，则在浏览窗口显示出虚拟计数器/定时器的图标。

(3) 将鼠标放置在 Counter Timer 之上，并使用快捷键“Ctrl+E”，打开编辑对话框进行设置。

(4) 选择需要的计数模式 (secs 或 hms)，以及 CE 和 RST 功能的逻辑极性。

(5) 退出编辑窗口，运行仿真。

3.3.4 虚拟终端 (Virtual Terminal)

PROTEUS VSM 提供的虚拟终端相当于键盘和屏幕的双重功能，免去了上位机系统的仿真模型，使用户在用单片机与上位机之间的串行通信时，直接由虚拟终端经 RS232 模型与单片机之间异步发送或接收数据。虚拟终端在运行仿真时会弹出一个仿真界面，当由 PC 向单片机发送数据时，可以和实际的键盘关联，用户可以从键盘经虚拟终端输入数据；

当接收到单片机发送来的数据后，虚拟终端相当于一个显示屏，会显示相应信息。

1. 虚拟终端的特性

- ☺ 全双工：以 ASCII 码的方式显示接收的串行数据，同时以 ASCII 码的方式传输键盘信号。
- ☺ 简单的两线串行数据接口：RXD 接收数据；TXD 发送数据。
- ☺ 简单的两线硬件握手方式：RTS 发送准备好；CTS 清除发送数据。
- ☺ 波特率范围为 300 ~ 57 600bps。
- ☺ 7 或 8 个数据位。
- ☺ 包含奇校验、偶校验，无校验。
- ☺ 具有 0、1 或 2 位停止位。
- ☺ 除硬件握手外，系统还提供了 XON/XOFF 软件握手方式。
- ☺ 可对 RX/TX 和 RTS/CTS 引脚输出极性不变或极性反向的信号。

2. 使用虚拟终端

单击工具箱中的 Virtual Instrument 按钮，则在对象选择窗口列出所包含的项目。

从对象选择器中选择“VIRTUAL TERMINAL”，则在浏览窗口显示出虚拟终端的图标。

在编辑窗口单击鼠标左键，添加虚拟终端，如图 3-125 所示。

- ☺ RXD 为数据接收端。
- ☺ TXD 为数据发送端。
- ☺ RTS 为请求发送信号。
- ☺ CTS 为清除传送，是对 RTS 的响应信号。

将虚拟终端的 RXD 和 TXD 引脚连接到待测系统的输出线和输入线上。RXD 是输入端，TXD 为输出端。

如果待测系统使用硬件握手方式，须将 RTS 和 CTS 引脚连接到数据流控制线上。RTS 为输出端，发信号，表明虚拟仪器已准备好接收数据。而 CTS 为输入信号，在虚拟终端发送数据前，这一信号必须为高（或浮动）。

选中虚拟仪器，并单击鼠标左键，弹出编辑对话框，出现如图 3-126 所示的虚拟终端属性设置对话框。

- ☺ Baud Rate: 波特率，范围为 300~57 600bps。
- ☺ Data Bits: 传输的数据位数，7 位或 8 位。
- ☺ Parity: 奇偶校验位，包括奇校验、偶校验和无校验。
- ☺ Stop Bits: 停止位，具有 0、1 或 2 位停止位。
- ☺ Send XON/XOFF: 第 9 位发送允许/禁止。

选择合适电路的波特率、数据长度、奇偶校验、流控制方式和极性设置。

设置完成后，开始仿真。在仿真界面中，按下运行按钮，将弹出虚拟终端仿真界面，如图 3-127 所示。

当其接收到数据后，会立即在终端显示；当传输数据到系统时，将光标置于虚拟终端屏幕，使用 PC 键盘输入数据。

仿真开始后，在虚拟终端屏幕单击鼠标右键，将弹出菜单选项，如图 3-128 所示。

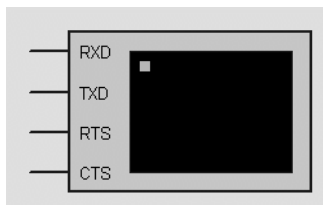


图 3-125 虚拟终端的原理图符号

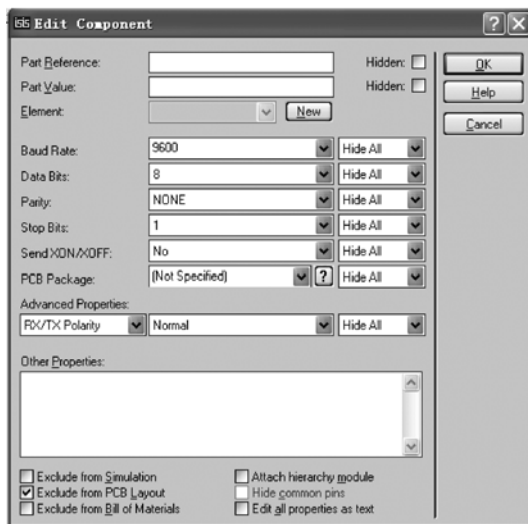


图 3-126 虚拟终端属性设置对话框

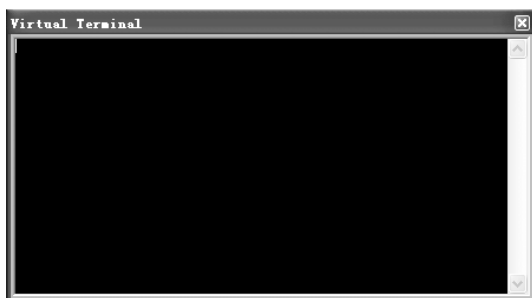


图 3-127 虚拟终端仿真界面

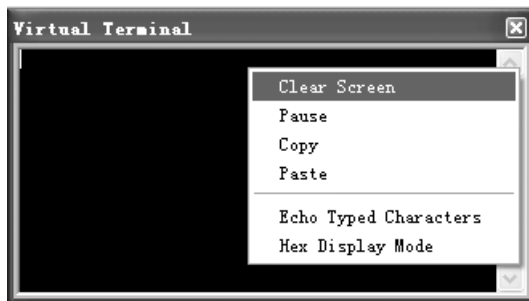


图 3-128 虚拟终端屏幕弹出式菜单

可根据需要选择相应的操作。这一菜单可实现：清屏、停止显示、复制或粘贴等操作，同时还可选择屏幕的显示方式。

- ⑤ 虚拟终端支持 ASCII 控制代码 CR (0Dh)、BS (0x08h) 和 BEL (0x07h)。而其他的代码，包括 LF (0x0A)，将被忽略。
- ⑤ 虚拟终端为纯数字模型，因此其引脚没有特殊电平要求。也就是说，可以将其直接连接到 CPU 或 UART，而不需要通过 RS232 的驱动器件（例如具有逻辑电平转换的 MAX232）。
- ⑤ RXD 和 TXD 引脚的默认值为高电平，因此静止状态为高电平，其起始位为逻辑低，而停止位为逻辑高。数据位中，逻辑高代表“1”，而逻辑低代表“0”。这与多数微控制器 UART 定义兼容，并且与诸如 6 850 和 8 250 的定义也兼容。当与上述不符时（如将虚拟终端连接到 RS232 驱动器件的输出端），用户需要设置 RXD/TXD 极性反向。
- ⑤ RTS 和 CTS 引脚的默认值也为高电平。如果希望将这些引脚连接到反向控制线（例如：RTS/CTS），用户需要设置 RTS/CTS 极性反向。
- ⑤ 在默认状况下，虚拟终端不显示用户输入的字符；也就是说，主系统将驱动输出终端显示用户输入的字符。如果用户希望显示输入的字符，须从右键的菜单中选择

Echo Typed Characters 选项。

- ☺ 使用 TEXT 属性预定义传输数据。这一功能使得电路在启动时就传输数据。例如：
在 TEXT 中输入：

TEXT="Hello World"

系统将会传输 “Hello World” 到电路。

【MAX232 模型】PROTEUS VSM 提供 RS232 驱动器件 MAX232，因此虚拟终端可以按照以下方式连接到目标 CPU，如图 3-129 所示。

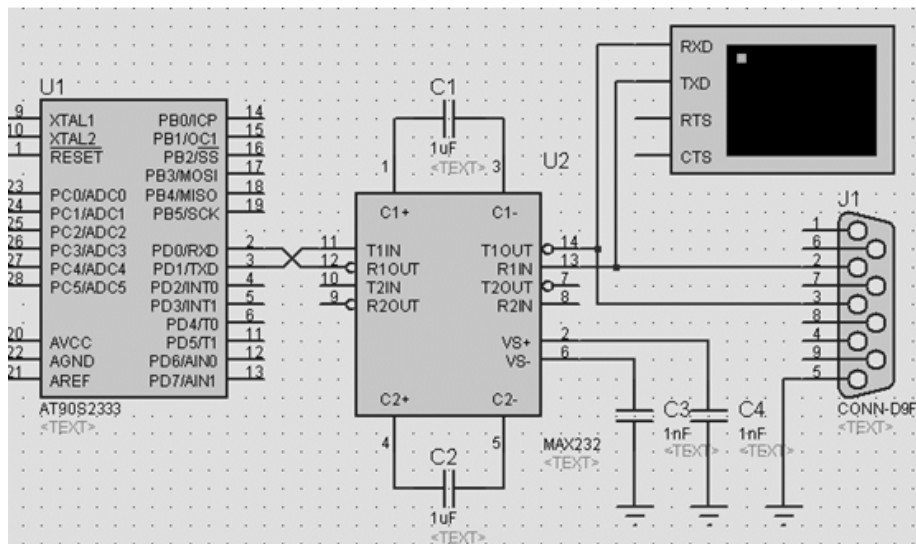


图 3-129 MAX232 与虚拟终端连接

注意：

MAX232 包含逻辑反相器，因此为保证电路的正确仿真，虚拟终端的 RXD/TXD 极性必须反转。

MAX232 数字模型不能进行内部电压转换操作的仿真。仿真这一特性可能带来巨大的性能损失。同样，如果在 TXDOut/RXDIn 引脚连接类似器件（电阻、电容、振荡器等），也会发生同样的仿真错误。

3.3.5 SPI 调试器（SPI Debugger）

SPI（Serial Peripheral Interface，串行设备接口）总线系统是 Motorola 公司提出的一种同步串行外设接口，允许 MCU 与各种外围设备以同步串行通信方式交换信息。其外围设备种类繁多，从简单的 TTL 移位寄存器到复杂的 LCD 显示驱动器、网络控制器等，可谓应有尽有。SPI 总线可直接与厂家生产的多种标准外围器件直接接口。

SPI Protocol Debugger（SPI 协议调试器）同时允许用户与 SPI 接口交互。这一调试器允许用户查看沿 SPI 总线发送的数据，同时也可以向总线发送数据。

图 3-130 所示为 SPI 调试器的原理图符号。其元件共有五个接线端，分别如下。

- ☺ DIN：接收数据端。

- ☺ DOUT：输入数据端。

- ☺ SCK: 连接总线时钟端。
- ☺ \overline{SS} : 从模式选择端, 从模式时必须为低电平才能使终端响应; 主模式时当数据正传输时此端为低电平。
- ☺ TRIG: 输入端, 能够把下一个存储序列放到 SPI 的输出序列中。双击 SPI 原理图符号, 可以打开它的属性设置对话框, 如图 3-131 所示。

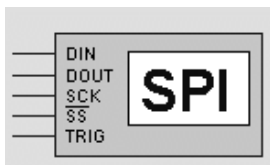


图 3-130 SPI 的原理图符号

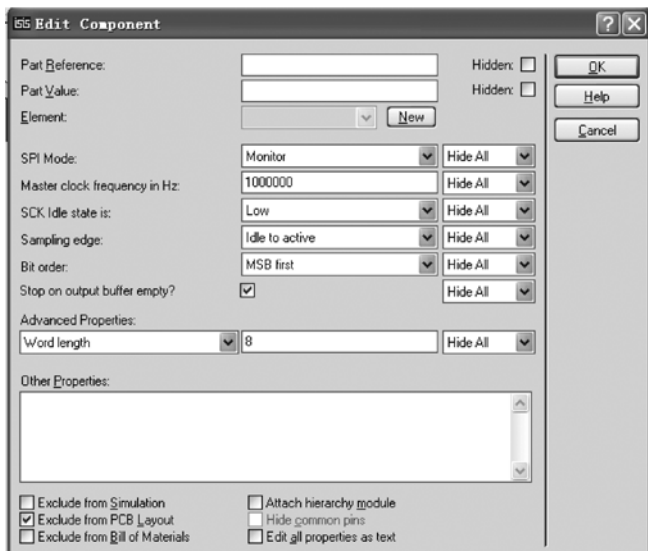


图 3-131 SPI 属性设置对话框

- ☞ SPI Mode: 有三种工作模式可以选择, Monitor 为监控模式, Master 为主要模式, Slave 为从模式。
- ☞ Master clock frequency in Hz: 主模式时钟频率 (Hz)。
- ☞ SCK Idle state is: SCK 空闲状态为高或低, 选择一个。
- ☞ Sampling edge: 采样边, 指定 DIN 引脚采样的边沿, 选择 SCK 从空闲到激活状态, 或从激活状态到空闲状态。
- ☞ Bit order: 位顺序, 指定一个传输数据的位顺序, 可先传送最高位 MSB, 也可先传送最低位 LSB。

1. 使用 SPI 调试器接收数据

- (1) 将 SCK 和 DIN 引脚连接到电路的相应引脚。
- (2) 将鼠标放置在 SPI 调试器之上, 并使用快捷键 “Ctrl+E”, 打开编辑对话框进行设置。
- (3) 设置 SPI 调试器字长、位顺序、SCK 空闲状态和采样边沿等属性。
- (4) 单击运行按钮, 开始仿真。此时, 将弹出调试器的弹出式窗口, 如图 3-132 所示。接收的数据将显示在窗口中。

2. 使用 SPI 调试器传输数据

- (1) 将 SCK 和 DIN 引脚连接到电路的相应引脚。

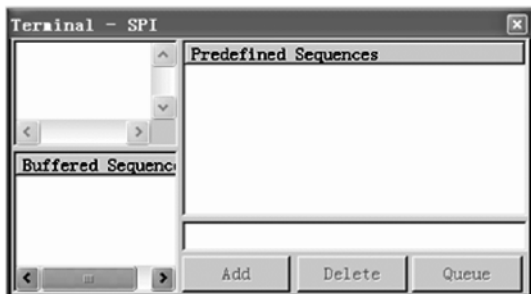


图 3-132 SPI 调试器弹出式窗口

(2) 将鼠标放置在 SPI 调试器之上, 并使用快捷键 “Ctrl+E”, 打开编辑对话框进行设置。

(3) 设置 SPI 调试器字长、位顺序、SCK 空闲状态和采样边沿等属性。

(4) 单击 “Pause” 按钮启动仿真, 并调出 SPI 调试器的弹出式窗口。

(5) 在弹出式窗口右下方的窗口中输入需要传输的数据, 如图 3-133 所示。

(6) 当输入需要传输的数据后, 即可直接传输数据, 也可以使用 “Add” 按钮, 将数据存放到 Predefined Sequences 列表中, 以备以后使用。

(7) 单击 “Play” 按钮, 在缓冲器列表中初始化传输项。

(8) 当仿真再次暂停时, 若序列输入窗口为空, 也可选择预定义序列, 并单击 “Queue” 按钮, 将任意预定义序列复制到缓冲器序列队列, 如图 3-134 所示。

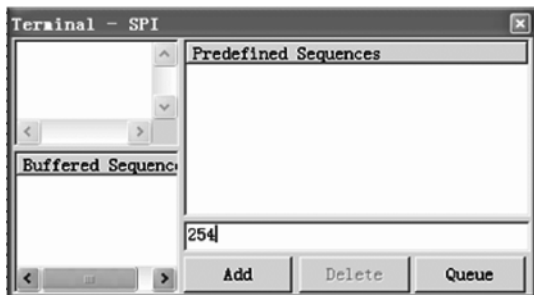


图 3-133 在 SPI 调试器弹出式窗口输入传输数据

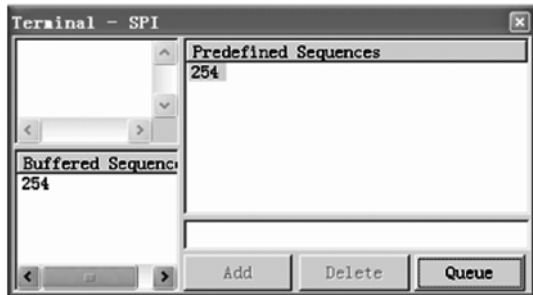


图 3-134 使用缓冲器序列队列传输数据

(9) 当再次激活仿真时, 这一序列将被传输。

3.3.6 I²C 调试器 (I²C Debugger)

I²C 总线简介。I²C (Intel IC) 总线是 Philips 公司推出的芯片间串行传输总线。它只需要两根线 (串行时钟线 SCL 和串行数据线 SDA), 就能实现总线上各器件的全双工同步数据传送, 可以极为方便地构成系统和外围器件扩展系统。I²C 总线采用器件地址的硬件设置方法, 避免了通过软件寻址器件片选线的方法, 使硬件系统的扩展简单灵活。按照 I²C 总线规范, 总线传输中的所有状态都生成相应的状态码, 系统的主机能够依照这些状态码自动地进行总线管理, 用户只要在程序中装入这些标准处理模块, 根据数据操作要求完成 I²C 总线的初始化, 启动 I²C 总线就能自动完成规定的数据传送操作。由于 I²C 总线接口已集成在片内, 用户无须设计接口, 使设计时间大为缩短, 且从系统中直接移去芯片对总线上的其他芯

片没有影响，这样方便产品的改性或升级。

虚拟仪器中的 I²C Debugger 允许用户检测 I²C 接口并与之交互，用户可以查看 I²C 总线发送的数据，同时也可以向总线发送数据。

I²C 调试器的原理图符号如图 3-135 所示。I²C 调试器共有三个接线端，分别如下。

- ☺ SDA：双向数据线。
- ☺ SCL：双向输入端，连接时钟。
- ☺ TRIG：触发输入，能引起存储序列被连续地放置到输出队列中。双击该元件，打开属性设置对话框，如图 3-136 所示。

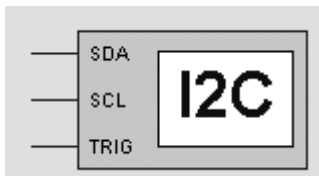


图 3-135 I²C 调试器的原理图符号

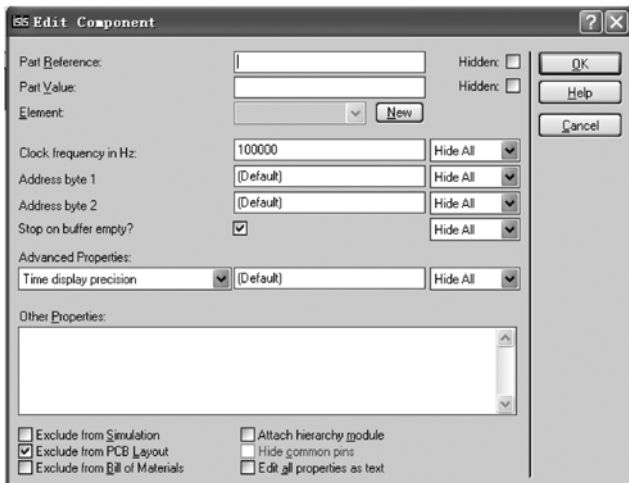


图 3-136 I²C 调试器属性设置对话框

☞ Address byte 1: 地址字节 1，如果使用此终端仿真一个从元件，则这一属性指定从器件的第一个地址字节。

☞ Address byte 2: 地址字节 2，如果使用此终端仿真一个从元件，并期望使用 10 位地址，则这一属性指定从器件的第二个地址字节。

I²C 调试器的仿真运行界面与 SPI 类似。

I²C 调试器具有许多用户可配置的属性（用于传输数据）。所有的属性都可通过编辑元件对话框（选中器件，单击鼠标左键，即可弹出编辑元件对话框）进行编辑。

- ☺ Address byte 1: 地址字节 1。如果用户使用这一终端仿真一个从器件，则这一属性用于指定 slave device 地址的第一个地址字节。主机使用最低有效位用于指示是否传输是一个读或写，或被忽略（for the purpose of addressing）。如果这一属性设置框为默认值，或为空，则这一终端将不被认作从器件。
- ☺ Address byte 2: 地址字节 2。如果用户使用这一终端仿真一个 slave device，并期望使用 10 位地址，则这一属性用于指定 slave device 地址的第二个地址字节。如果这一属性设置框为空，则假定地址为 7 位。
- ☺ Stop on buffer empty: 为空时停止。指定当输出缓冲器为空，并且一个字节要求被发送时，是否仿真停止。
- ☺ Advanced Properties: 在这一设置中允许用户指定预先存放输出序列的文本文件的名称。如果这一属性设置为空，则序列作为器件属性的一部分进行保存。

除以上属性外，当接收数据时，I²C 终端需要使用一种特殊的序列句法。这一句法出现在输入数据显示窗口（调试器窗口左上方），包括序列起始（sequence starts）和（acknowledges）。显示的序列字符如下：

- ☺ S Start Sequence.
- ☺ Sr Restart Sequence.
- ☺ P Stop Sequence.
- ☺ N Negative Acknowledge received.
- ☺ A Acknowledge received.

3.3.7 信号发生器（Signal Generator）

PROTEUS 的虚拟信号发生器主要有以下功能：

- ☺ 产生方波、锯齿波、三角波和正弦波；
- ☺ 输出频率范围为 0~12MHz，8 个可调范围；
- ☺ 输出幅值为 0~12V，4 个可调范围；
- ☺ 幅值和频率的调制输入和输出。

信号发生器的原理图符号如图 3-137 所示。

信号发生器有两大功能，一是输出非调制波，二是输出调制波。通常使用它的输出非调制波功能来产生正弦波、三角波和锯齿波，方波直接使用专用的脉冲发生器来产生比较方便，主要用于数字电路中。

在用于非调制波发生器时，信号发生器的下面两个接头“AM”和“FM”悬空不接，右面两个接头“+”端接至电路的信号输入端，“-”端接地。

仿真运行后，出现如图 3-138 所示的界面。

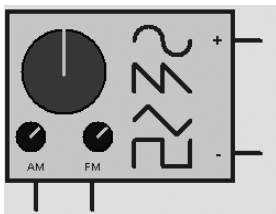


图 3-137 信号发生器原理图符号

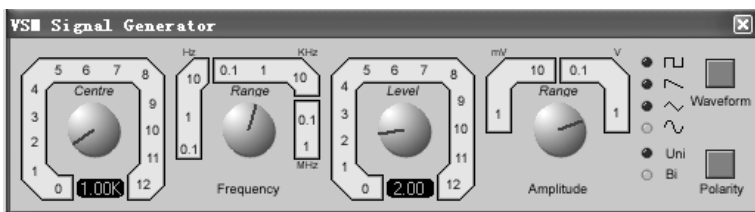


图 3-138 信号发生器仿真运行后的界面

最右端两个方形的按钮，上面一个用来选择波形，下面一个选择信号电路的极性，即是双极型（Bi）还是单极型（Uni）三极管电路，以和外电路匹配。最左边两个旋钮用来选择信号频率，左边是微调，右边是粗调。中间两个旋钮用来选择信号的幅值，左边是微调，右边是粗调。如果在运行过程中关掉信号发生器，则需要从主菜单 Debug 中选取最下面的 VSN Signal Generator 来重现。

设置频率度盘，以满足应用电路的需求。当 Centre 的指针被设置在 1 的位置时，Range 值表明所发生信号的频率。

设置幅度度盘，以满足应用电路的需求。当 Level 的指针被设置在 1 的位置时，Range 值表明所发生信号的幅值。幅值为输出电平的峰-峰值。

单击“Waveform”按钮，代表波形类型的 LED 灯将会点亮，从而选择适合电路的输出

信号。

使用 AM & FM 调制输入。信号发生器模型支持调幅波和调频波的输出。幅值输入和频率输入具有以下特性：

- ☺ 调制输入的增益由 Frequency Range 和 Amplitude Range 分别按照 Hz/V 和 V/V 进行设置。
- ☺ 调制输入的电压范围为 $-12 \sim +12\text{V}$ 。
- ☺ 调制输入的输入阻抗为无穷大。
- ☺ 调制输入的电压值为 Range 设置值与 Centre/Level 设置值的乘积，倍乘后的值为幅度的瞬时输出频率。

例如，如果 Frequency Range 设置为 1kHz，同时 Frequency Centre 设置为 2.0，则 2V 的调频信号的输出频率为 4kHz。

3.3.8 模式发生器 (Pattern Generator)

模式发生器是模拟信号发生器的数字等价物，它支持 8 位 1KB 的模式信号，同时具有以下特征：

- ☺ 既可以在基于图表的仿真中使用，也可以在交互式仿真中使用；
- ☺ 支持内部和外部时钟模式及触发模式；
- ☺ 使用游标调整时钟刻度盘或触发器刻度盘；
- ☺ 十六进制或十进制栅格显示模式；
- ☺ 在需要高精度设置时，可直接输入指定的值；
- ☺ 可以加载或保存模式脚本文件；
- ☺ 可单步执行；
- ☺ 可实时显示工具包；
- ☺ 可使用外部控制，使其保持当前状态；
- ☺ 栅格上的块编辑命令使得模式配置更容易。

【模式发生器的原理图符号及引脚说明】模式发生器的原理图符号如图 3-139 所示，各接线端含义如下。

- ☺ CLKIN: 外部时钟信号输入端，系统提供两种外部时钟模式。
- ☺ HOLD: 外部输入信号，用来保持模式发生器目前状态，高电平有效。
- ☺ TRIG: 触发输入端，用于将外部触发脉冲信号反馈到模式发生器。系统提供五种外部触发模式。
- ☺ OE: 输出使能信号输入端，高电平有效，模式发生器可输出模式信号。
- ☺ CLKOUT: 时钟输出端，当模式发生器使用的是外部时钟时，可以用于镜像内部时钟脉冲。
- ☺ CASCADE: 级联输出端，用于模式发生器的级联，当模式发生器的第一位被驱动，并且保持高电平时，此端输出高电平，保持到下一位被驱动之后一个周期时间。
- ☺ B[0..7] 和 Q0 ~ Q7: 分别为数据输入和输出端。

【模式发生器的属性设置对话框主要参数说明】双击模式发生器的原理图符号，则弹出其属性对话框，如图 3-140 所示。

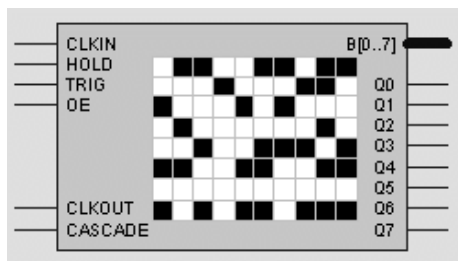


图 3-139 模式发生器原理图符号

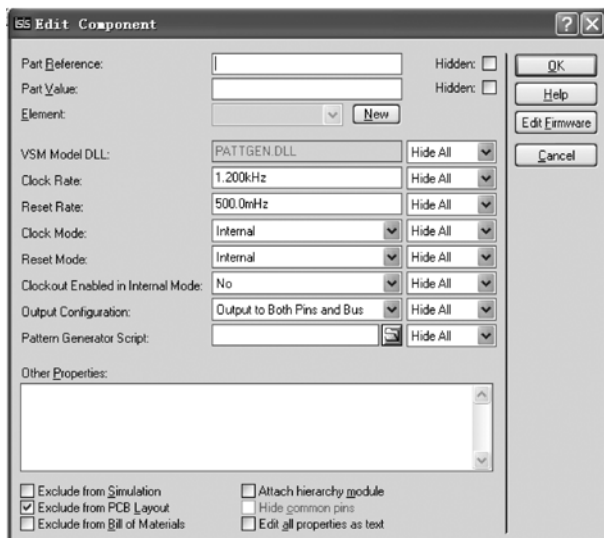


图 3-140 模式发生器属性设置对话框

- ☺ Clock Rate: 时钟频率。
- ☺ Reset Rate: 复位频率。
- ☺ Clock Mode: 时钟模式, 有以下三种。
 - ☞ Internal: 内部时钟;
 - ☞ External Pos Edge: 外部上升沿时钟;
 - ☞ External Neg Edge: 外部下降沿时钟。
- ☺ Reset Mode: 复位模式, 有以下四种。
 - ☞ Internal: 内部复位;
 - ☞ Async External Pos Edge: 异步外部上升沿脉冲;
 - ☞ Async External Neg Edge: 同步外部上升沿脉冲;
 - ☞ Sync External Neg Edge: 异步外部下降沿脉冲。
- ☺ Clockout Enabled in Internal Mode: 内部模式下时钟输出使能。
- ☺ Output Configuration: 输出配置, 共三种。
 - ☞ Output to Both Pins and Bus: 引脚和总线均输出;
 - ☞ Output to Pins Only: 仅在引脚输出;
 - ☞ Output to Bus Only: 仅在总线输出。
- ☺ Output Generator Script: 模式发生器脚本文件。

参数设置完成后, 单击“OK”按钮结束。

在仿真界面中, 单击运行按钮, 将弹出模式发生器编辑窗口, 如图 3-141 所示。

通过左击模式发生器中的栅格并设置其逻辑状态, 设置用户需要发生的模式。

确定时钟是内部时钟还是外部时钟, 然后设置时钟模式按钮。

- ☺ 如果使用的是内部时钟, 可通过调整时钟刻度盘以期得到期望的时钟频率。

确定触发是内部触发还是外部触发, 然后使用触发按钮设置相应的模式。如果是外部触发, 用户需要考虑是同步时钟触发还是异步时钟触发。

- ☺ 如果是内部触发, 可通过调整触发刻度盘以期得到期望的触发频率。

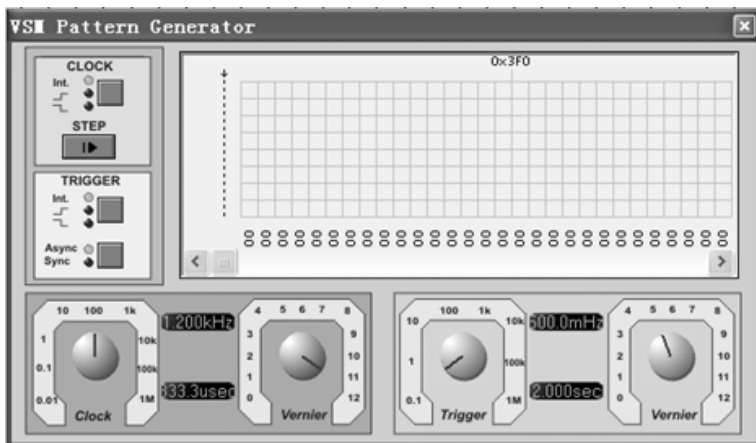


图 3-141 模式发生器编辑窗口

单击仿真界面中的运行按钮，即可输出相应的模式。

- ☺ 如果期望得到单时钟周期的模式信号，须在仿真控制面板中单击“Step”按钮，使栅格向左移动。

1. 在基于图表的仿真中使用模式发生器输出模式

- (1) 创建电路。
- (2) 在原理图中用户感兴趣的部位插入探针，并将这些探针添加到图表。
- (3) 在原理图中选中模式发生器，并单击鼠标左键，打开元件编辑对话框。
- (4) 根据系统要求，配置触发选项和时钟选项。
- (5) 在模式发生器脚本区域（Pattern Generator Script Field）加载期望的模式文件。
- (6) 退出模式发生器编辑对话框，并单击空格“Space”按钮，运行仿真。

2. 模式发生器元件引脚

- ☺ 数据输出引脚（Tri-State Output）。模式发生器可以在总线上输出，也可在单个引脚上输出。
- ☺ 时钟输出引脚 Clock-Out Pin（Output）。当模式发生器使用的是外部时钟时，用户可以配置这一引脚，用以镜像内部时钟脉冲。这一功能是模式发生器的一个属性，可以通过编辑元件对话框进行修改。在默认情况下，这一选项是无效的，因为它可能导致仿真故障，尤其在高频时钟仿真时更容易引发这样的恶性事件。
- ☺ Cascade Pin（Output）。当模式的第一位被驱动，并且保持高电平时，层叠引脚 Cascade Pin 被拉高，直至模式的下一位被驱动（一个时钟周期之后）。这就意味着当开始仿真的第一个时钟周期，Cascade Pin 为高电平，而在第一个时钟周期之后被复位。
- ☺ 触发引脚 Trigger Pin（Input）。这一引脚用于将外部触发脉冲信号反馈到模式发生器。系统提供四种外部触发模式。
- ☺ 时钟输入引脚 Clock-In Pin（Input）。这一引脚用于输入外部时钟信号。系统提供两种外部时钟模式。

- ☺ 保持引脚 Hold Pin (Input)。这一引脚为高电平期间，模式发生器将保持在暂停点，直至这一引脚被释放。对于内部时钟或内部触发，时钟将从暂停点重新开始。例如，对于 1Hz 的内部时钟，如果模式发生器暂停在 3.6s 处，在 5.2s 处重新开始，则下一个下降时钟边沿将发生在 5.6s 处。
- ☺ 输出使能引脚 Output Enable Pin (Input)。使用高电平驱动这一引脚，则模式发生器可输出模式信号。如果这一引脚不为高电平，虽然模式发生器依然按照指定的模式运行，但并不驱动模式输出引脚输出模式信号。

3. 时钟模式 (Clocking Modes)

☺ 内部时钟 (Internal Clocking)

☞ 内部时钟是负沿脉冲。

☞ 内部时钟既可在仿真之前使用元件编辑对话框进行指定，也可在仿真暂停期间使用时钟模式按钮进行指定。

☞ 当时钟输出引脚 Clock-Out Pin 被激活时，可镜像内部时钟。在默认状况下，这一选项是无效的，因为它可能导致仿真故障，尤其在高频时钟仿真时更容易引发恶性事件。但是，如果需使用这一引脚，可以使用模式发生器的编辑元件对话框来激活。

☺ 外部时钟 (External Clocking)

☞ 有两种外部时钟模式：负沿脉冲 (low-high-low) 和正沿脉冲 (high-low-high)。

☞ 使用外部连线，将外部时钟脉冲连接到时钟输入引脚 Clock-In，并选择一种时钟模式。

☞ 在外部时钟模式下，可以通过在仿真前编辑元件或仿真暂停期间改变外部时钟模式。

4. 触发模式 (Trigger Modes)

【内部触发】 模式发生器的内部触发模式按照指定的间隔触发。如果时钟是内部时钟，则时钟脉冲在这一触发点复位，如图 3-142 所示。

例如，设定内部时钟为 1Hz，并且设定内部触发时间为 3.75s，则层叠引脚 (Cascade Pin) 在模式第一位时为高，而在其他时间为低。

📖 注意：

在触发时间，内部时钟同时被复位。模式的第一位被输出。层叠引脚被拉高。

【外部异步正脉冲触发】 触发器在触发引脚由正边沿转换指定。当触发发生时，触发器立即动作，下一个时钟边沿将要在 bitclock/2，与复位并发由低到高转换，如图 3-143 所示。

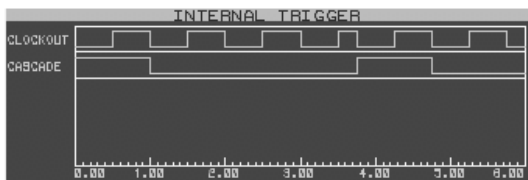


图 3-142 内部触发

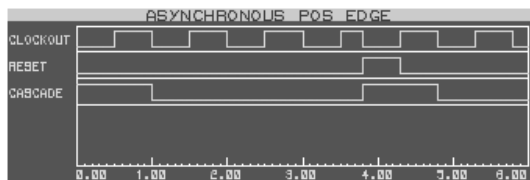


图 3-143 外部异步正脉冲触发

例如，设定内部时钟为 1Hz，并且触发引脚在 3.75s 时拉高，则立即在触发引脚的正边沿沿时钟复位，模式的第一位驱动层叠引脚。

【外部同步正脉冲触发】触发器在触发引脚由正边沿转换指定。触发被锁定，和下一个时钟的下降沿同步动作，如图 3-144 所示。例如，设定内部时钟为 1Hz。

注意：

时钟不受触发影响，并且触发器在时钟下降沿动作，与正边沿脉冲并发。

【外部异步负脉冲触发器】触发器在触发引脚由负边沿转换指定。当触发发生时，触发器立即动作，且模式的第一位在输出引脚输出，如图 3-145 所示。



图 3-144 外部同步正脉冲触发

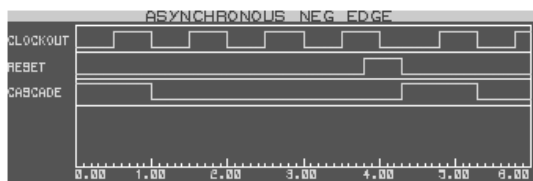


图 3-145 外部异步负脉冲触发

例如，设定内部时钟为 1Hz。从图中可以看到时钟在触发脉冲的负边沿复位，并且模式的第一位在那时被驱动。

5. 外部同步负脉冲触发器

触发器在触发引脚由负边沿转换指定。触发被锁定，和下一个时钟的下降沿同步动作，如图 3-146 所示。例如，设定内部时钟为 1Hz。

注意：

触发发生在触发脉冲的下降沿，模式直到时钟脉冲的下降沿，与触发动作并发复位。

6. 外部保持 (External Hold)

保持模式发生器的当前状态。如果想要在一段时间内保持模式，可以在期望保持的那段时间内使保持引脚 (Hold Pin) 为高电平，如图 3-147 所示。

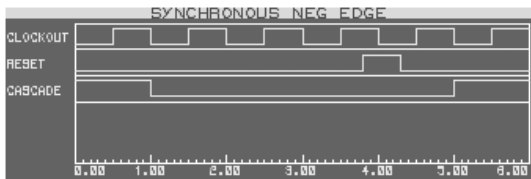


图 3-146 外部同步负脉冲触发

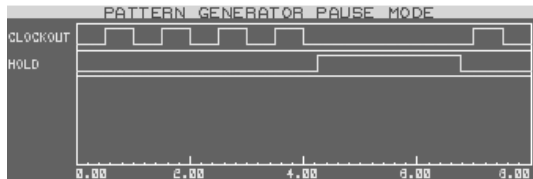


图 3-147 外部保持

如果使用的是内部时钟，在释放保持引脚的同时，模式发生器将重新启动。也就是说，保持引脚在时钟周期的一半变高。然后，当释放保持引脚时，下一位将要在以后的时钟周期的一半时驱动输出引脚。

若保持引脚为高，则内部时钟被暂停。当释放保持引脚时，时钟将在相对于一个时钟

周期的暂停点重新启动。

7. 附加功能 (Additional Functionality)

【加载和保存模式脚本】在栅格上单击鼠标右键，将弹出菜单选项，如图 3-148 所示。

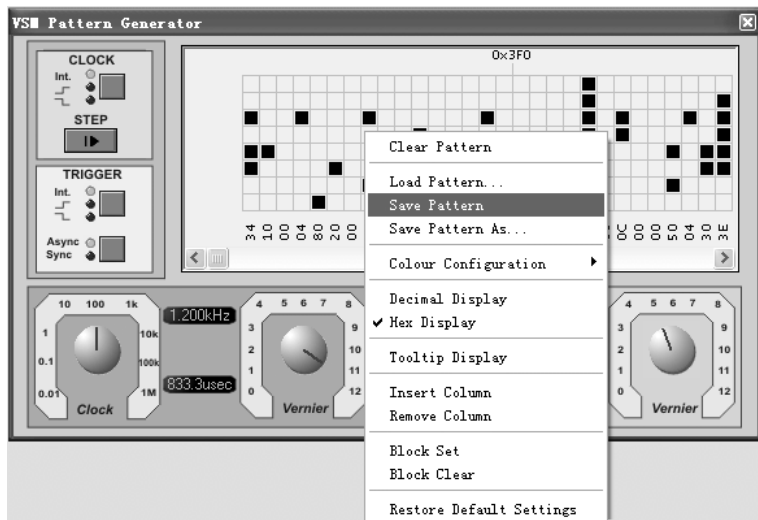


图 3-148 右键菜单选项

从弹出的菜单中选择相关的选项，则可加载或保存模式脚本。如果用户想要在多个设计中使用特殊模式，这一方法是非常有用的。

模式脚本为纯文本文件，每个字节由逗号分隔。每个字节代表栅格上的一栏。以分号起始的行被记为注释行，并且被剖析器忽略。在默认情况下，字节格式为十六进制。当用户创建脚本文件时，输入值可以是十进制、二进制或十六进制数。

【为刻度盘设置指定值】用户可以通过双击合适的刻度盘，指定位和触发频率的精确值。双击后，将出现浮动的编辑框，如图 3-149 所示。

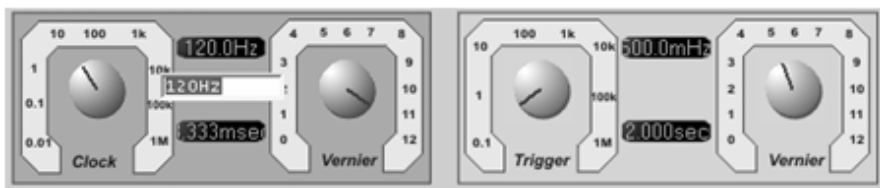


图 3-149 使用浮动编辑框设置刻度盘

用户可以将设置值输入其中。在默认情况下，输入值被认作频率，同时，用户也可以通过为输入值加上合适的后缀如 (sec、ms 等) 来指定输入值的类型。此外，如果用户希望触发为时钟的精确倍乘，可以附加期望的倍乘后缀 (如 5bits)。

按 Enter 键或 Escape 键，或单击模式发生器窗口的任何其他部位，用于确认输入。

说明:

通过对编辑元件对话框的合理设置，可以指定这些值的周期，以便进行仿真。

【设定模式栅格的值】在显示当前那一栏的值的文本上单击鼠标左键，可以指定栅格上该栏中任何一个值。单击后，将出现一个浮动的编辑框，如图 3-150 所示。

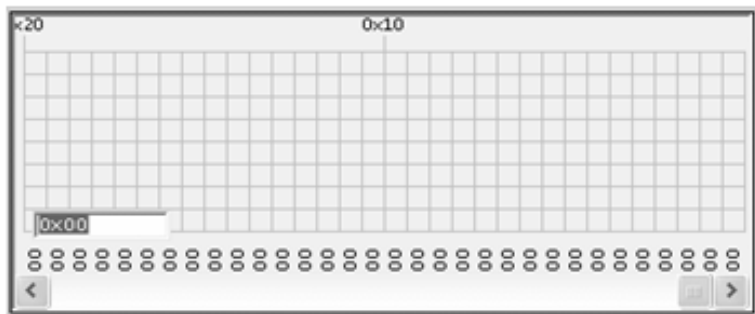


图 3-150 使用浮动编辑栅格

可在框中输入期望的值。用户可以输入十进制值（如 135）、十六进制值（如 0xA7），或二进制值（如 0b10110101）。

按 Enter 键或 Escape 键，或单击模式发生器窗口的任何其他部位，用于确认输入。

说明：

为了方便，用户可以在想要编辑的栏，使用快捷键“Ctrl+I”设置一栏，而使用“Ctrl+Shift+I”清除一栏。

8. 手动指定模式的周期长度

在期望模式结束的栏的栅格上单击鼠标左键，可以手动指定周期。在同样的区域单击鼠标右键，可以取消周期。

- ☺ 当时间周期与内部时钟或外部时钟所指定的时钟相等时，单步执行按钮可用于进一步仿真。直到下一个时钟周期完成，并且再次单击单步执行按钮，仿真将继续。
- ☺ 栅格的显示方式可在十六进制和十进制显示方式间切换。在栅格上右击鼠标，从菜单中选择期望的选项，或使用快捷键“Ctrl+X”（十六进制显示）或“Ctrl+D”（十进制显示）切换，可实现显示模式的转换。
- ☺ 单击模式发生器的原理图部分，以产生编辑元件对话框。实际上属性允许用户配置是否在总线和引脚上均输出模式，或只在总线，或只在引脚输出模式。
- ☺ 用户可设定随鼠标移动而显示当前行、当前列信息的工具栏的可用性。通过单击鼠标右键出现的文本，或快捷键“Ctrl+Q”实现对工具栏的控制。

注意：

在 Block Set 或 Block Clear 期间，工具栏是不可用的。

- ☺ 用户可使用 Block Set 和 Block Clear 命令实现对栅格期望模式的配置。通过单击鼠标右键出现的文本，或快捷键“Ctrl+S”和“Ctrl+C”实现上述操作。

注意：

块编辑命令（Block Editing Commands）在 tooltip 模式是不可用的。

3.3.9 电压表和电流表 (AC/DC Voltmeter/Ammeter)

PROTEUS VSM 提供了四种电表，分别是 AC Voltmeter（交流电压表）、AC Ammeter（交流电流表）、DC Voltmeter（直流电压表）和 DC Ammeter（直流电流表）。

(1) 在 PROTEUS ISIS 的界面中选择虚拟仪器图标，在出现的元件列表中，分别把上述四种电表放置在原理图编辑区域中，如图 3-151 所示。

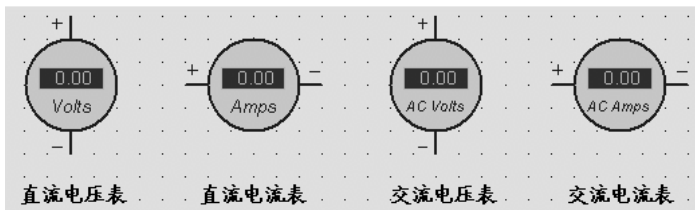


图 3-151 四种电表的原理图符号

(2) 双击任一电表的原理图符号，出现其属性对话框，图 3-152 所示是直流电流表的属性设置对话框。

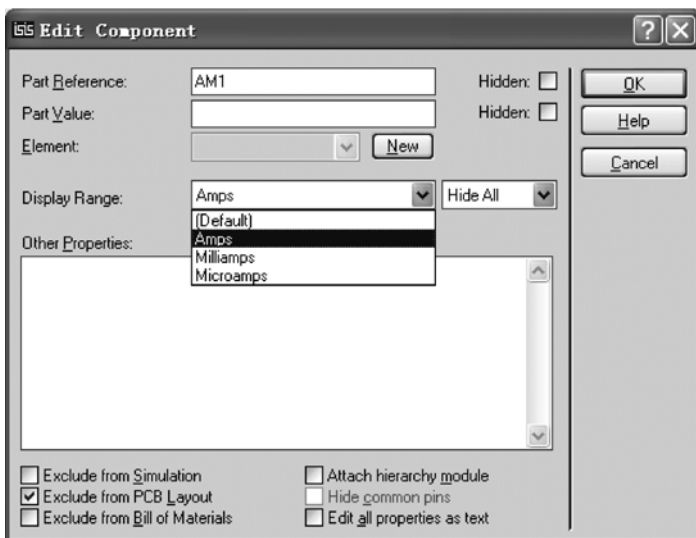


图 3-152 直流电流表的属性设置对话框

在元件名称 Part Reference 项中将该交流电压表命名为 AM1，元件值 Part Value 项中不填。在显示范围 Display Range 中有四个选项，用来设置该直流电流表是安培表 (Amps)、毫安表 (Milliamps) 或是微安表 (Microamps)，默认是安培表。单击“OK”按钮完成设置。

- ☺ 电表的 FSD 的分辨率为三位，最大可显示两位小数位。同时，通过编辑元件，选择需要的 Display Range 属性，可设置显示范围：100、100m 和 100μ。
- ☺ 电压表模型支持内置电阻 (internal load resistance) 属性，其默认值为 100M。同时，可以通过编辑元件改变这一值。当 internal load resistance 为空时，即加载模型内阻选项无效。

- ☺ 交流电压表和交流电流表显示的是在用户定义的时间常量内电压或电流的平均值 RMS。




3.4 探针

探针用于记录所连接网络的状态，ISIS 系统提供了两种探针，即电压探针（Voltage Probe）和电流探针（Current Probe）。

- ☺ 电压探针：既可以在模拟仿真中使用，也可以在数字仿真中使用。在模拟电路中记录真实的电压值，而在数字电路中，记录逻辑电平及强度。
- ☺ 电流探针：仅在模拟电路中使用，可显示电流方向。
- ☺ 探针既可用于图表的仿真，也可用于交互式仿真。

3.4.1 电压探针

(1) 单击工具箱中的 Probe Mode 按钮，在对象选择器中选择 VOLTAGE，将在预览窗口显示电压探针的图标。电压探针和原理图中其他元件一样，可以进行相关操作，如放置、编辑、旋转和移动等。

(2) 在编辑窗口单击，则电压探针被添加到原理图中，如图 3-153 所示。

当探针未被连接到任何已存在的导线上时，它默认名称为“？”，说明此时其未被标注。而当探针被连接到一个网络（如：探针被直接置于已存在的导线上），它将以这个网络名称作为标志。如果其所连接的网络也未被标注，则以其最接近的元件参考号或引脚名称作为标志。当其连线被断开，或被拖到其他网络时，探针的标志将随时被更新。

用户也可根据自己的要求，使用探针编辑对话框，编辑探针标志，此时，编辑后的探针标志将作为永久性表示，而不再被更新。电压探针编辑对话框如图 3-154 所示。

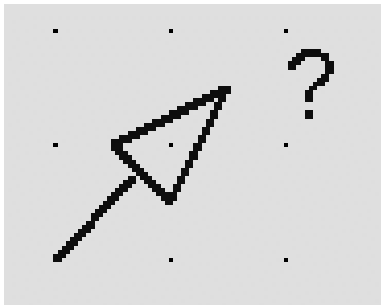


图 3-153 电压探针

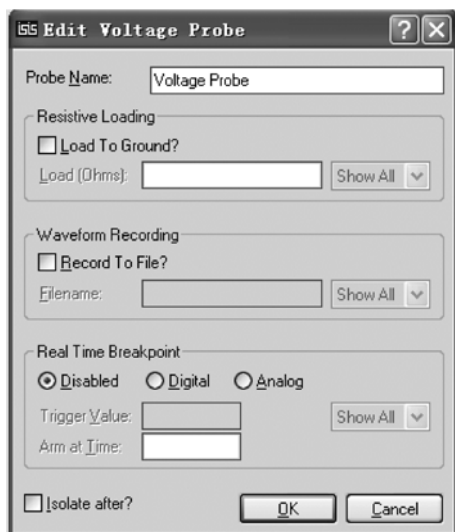


图 3-154 电压探针编辑对话框


电压探针编辑对话框包含两个设计项目。

- ☺ Load Resistance: 电压探针阻抗。当测量点与地之间没有直流通道时, 需设置电压探针的阻抗。
- ☺ Record Filename: 记录文件名称。电流或电压探针可以将数据记录到文件, 用于在 Tape 发生器中播放。探针的这一特性使得用户可以使用某一电路创建测试波形, 然后在另一电路中使用。

按照电路要求进行设置, 设置完成后, 单击“OK”按钮, 探针编辑完成。

3.4.2 电流探针

1. 电流探针的放置

(1) 单击工具箱中的 Probe Mode 按钮, 在对象选择器中选择 CURRENT, 将在预览窗口显示电流探针的图标。

(2) 电流探针和原理图中其他元件一样, 可以进行相关操作, 如放置、编辑、旋转和移动等。

(3) 在编辑窗口单击, 则电流探针被添加到原理图中, 如图 3-155 所示。

当探针未被连接到任何已存在的导线上时, 它默认名称为“?”, 说明此时其未被标注。而当探针被连接到一个网络(如: 探针被直接置于已存在的导线上), 它将以这个网络名称作为标志。如果其所连接的网络也未被标注, 则以其最接近的元件参考号或引脚名称作为标志。当其连线被断开, 或被拖到其他网络时, 探针的标志将随时被更新。

用户也可根据自己的要求, 使用探针编辑对话框, 编辑探针标志, 此时, 编辑后的探针标志将作为永久性表示, 而不再被更新。

电流探针有方向性, 需考虑回路中电流的流向。调整电流探针的方向与其他元件一样。



注意:

电流探针的方向为小圆圈内的箭头方向。

2. 电流探针的编辑

(1) 右击选中电流探针, 再单击, 进入电流探针编辑对话框, 如图 3-156 所示。

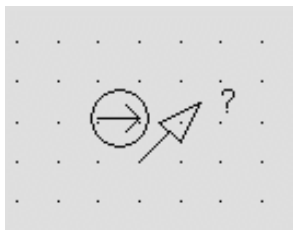


图 3-155 电流探针

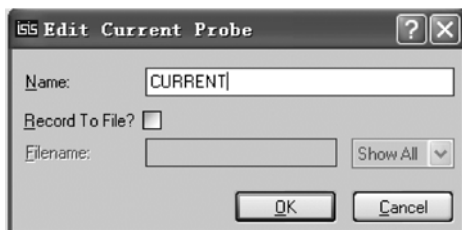


图 3-156 电流探针编辑对话框

电流探针编辑对话框包含的设计项目有 Record Filename, 即记录文件名称。电流或电压探针可以将数据记录到文件中, 用于在 Tape 发生器中播放。探针的这一特性使得用户可以使用某一电路创建测试波形, 然后在另一电路中使用。

(2) 按照电路要求进行设置, 设置完成后, 单击“OK”按钮, 则探针编辑完成。

第4章 PROTEUS ISIS 单片机仿真

在基于微处理器系统的设计中，即使没有物理原型，PROTEUS VSM 也能够进行软件开发。模型库中包含 LCD 显示、键盘、按钮、开关等通用外围设备。同时，提供的 CPU 模型有 ARM7、PIC、Atmel AVR、Motorola HCXX 以及 8051/8052 系列。

单片机系统的仿真是 PROTEUS VSM 的一大特色。同时，本仿真系统将源代码的编辑和编译整合到同一设计环境中，这样使得用户可以在设计中直接编辑代码，并可容易地查看用户修改源程序后对仿真结果的影响。本章以 PWM 输出控制电路为例，说明 PROTEUS ISIS 单片机仿真的过程。



4.1 在 PROTEUS ISIS 中输入单片机系统电路

在传统控制中，通常采用模拟量来控制被测对象，这样硬件较复杂且成本较高。而采用脉冲宽度调制（PWM）方法取代模拟量控制，采用继电器作为执行元件即可实现系统控制。数字脉宽调节常用的方法是脉冲周期固定不变，脉冲宽度可调。通过改变脉冲的宽度，就能达到改变占空比 τ/T 的目的，从而控制继电器的接通与断开，实现功率控制。

PWM 是单片机上常用的模拟量输出方法，通过外接的转换电路，可以将脉冲的占空比变成电压。程序中通过调整占空比来调节输出模拟电压。占空比是指脉冲中高电平与低电平的宽度比。

4.1.1 选取仿真元件

点选“Component”图标，单击“P”按钮，从弹出的选取元件对话框中选择 PWM 输出控制电路仿真元件。仿真元件信息如表 4-1 所示。

表 4-1 仿真元件信息（PWM 输出控制电路仿真）

元 件 名 称	所 属 类	所 属 子 类
AT89C51（51 系列单片机）	Microprocessor ICs	8051 Family
CAP（电容）	Capacitors	Generic
CAP-POL（电解电容）	Capacitors	Generic
CRYSTAL（晶振）	Miscellaneous	—
RES（电阻）	Resistors	Generic
POT-HG（滑动变阻器）	Resistors	Variable
ADC0808（模数转换）	Data Converters	AD Converters

将仿真元件添加到对象选择器后关闭元件选取对话框。

4.1.2 调试 PWM 输出电路中的 ADC0808 模数转换电路

1. ADC0808 模数转换器元件简介

ADC0808 模数转换器元件外观如图 4-1 所示。其引脚功能如下：

- ☺ IN0 ~ IN7: 八路模拟量输入。
- ☺ ADD A、ADD B、ADD C: 三位地址输入，三个地址输入端的不同组合选择八路模拟量输入。
- ☺ ALE: 地址锁存启动信号，在 ALE 的上升沿，将 ADD A、ADD B、ADD C 上的通道地址锁存到内部的地址锁存器。
- ☺ OUT1 ~ OUT8: 八位数据输出线，AD 转换结果由这八根线传送给单片机。
- ☺ OE: 允许输出信号。当 OE=1 时，即为高电平，允许输出锁存器输出数据。
- ☺ START: 启动信号输入端，START 为正脉冲，其上升沿清除 ADC0808 的内部各寄存器，其下降沿启动 AD 开始转换。
- ☺ EOC: 转换完成信号，当 EOC 上升为高电平时，表明内部 AD 转换已完成。
- ☺ CLOCK: 时钟输入信号。
- ☺ VREF(+), VREF(-): 基准电压。

使用 ADC0808 将外接模拟输入信号转换为数字信号，电路连接如图 4-2 所示。

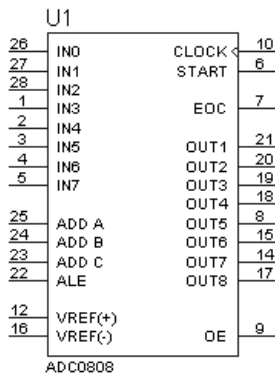


图 4-1 ADC0808 模数转换器元件外观

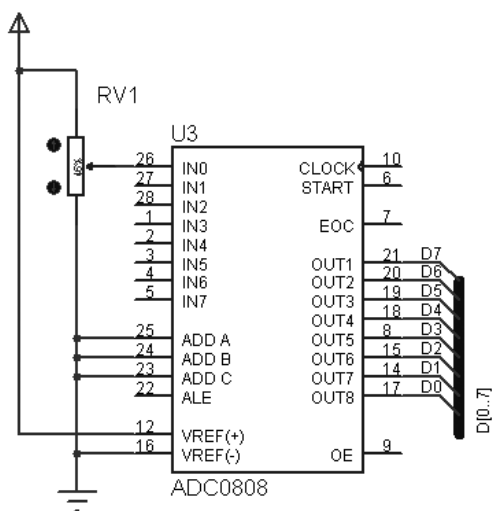


图 4-2 使用 ADC0808 模数转换电路

本电路使用 IN0 作为信号输入引脚，使用滑动变阻器作为模拟信号的输入端。滑动变阻器的设置如图 4-3 所示。变阻器在滑动过程中以线性方式变化。

2. 分析模数转换电路

模数转换电路仿真图如图 4-4 所示。

其中地址锁存信号设置如图 4-5 所示。地址锁存信号设置为脉冲信号，脉宽为 0.5s，为正向脉冲。系统时钟引脚接时钟信号源，时钟信号源设置如图 4-6 所示。

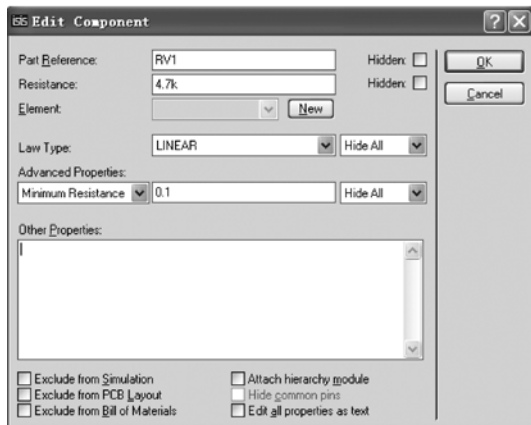


图 4-3 滑动变阻器的设置

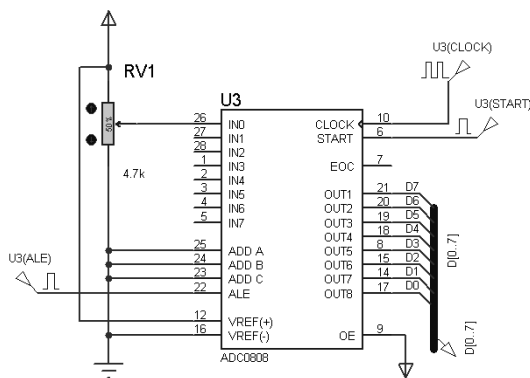


图 4-4 模数转换电路仿真图

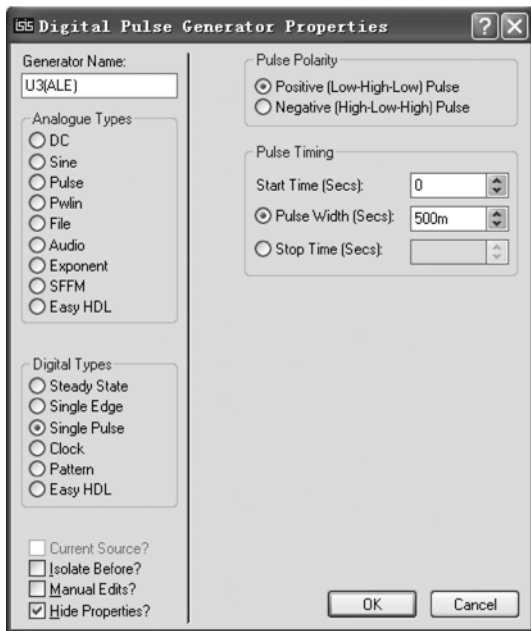


图 4-5 地址锁存信号设置

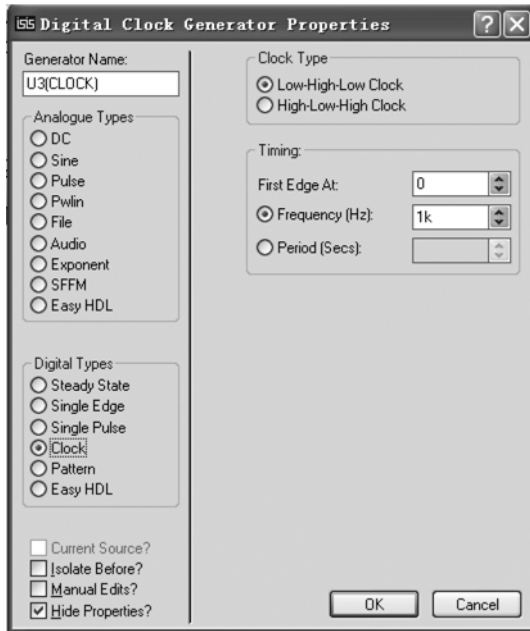


图 4-6 时钟信号源设置

时钟信号设置为频率 1kHz 的方波信号。AD 转换启动信号引脚的信号源设置如图 4-7 所示。

AD 转换启动信号设置为起始时刻为 1s、脉宽为 0.5s 的正向脉冲。允许输出信号端直接与电源相连。在电路总线添加电压探针，电压探针将自动被命名为 D[0..7]。在电路中添加数字仿真图表，并添加信号源，如图 4-8 所示。双击数字分析图表，设置仿真时间为 3s。设置滑动变阻器为 100%，如图 4-9 所示。

将鼠标放置到数字分析图表中，按空格键，仿真电路，结果如图 4-10 所示。从仿真图可知，启动 AD 转换后，系统将输入的模拟信号转换为数字量。

在电路的模拟信号输入端口连接电压表，如图 4-11 所示。

单击电路中的运行按钮，仿真电路，结果如图 4-12 所示。从仿真结果可知，当输入电

压为 5V 时, 电路模数转换后的结果为 0FFH, 则当电路输入模拟电压为 2.5V 时, 输入数字量为 $256/2-1=127$, 即为 7FH。

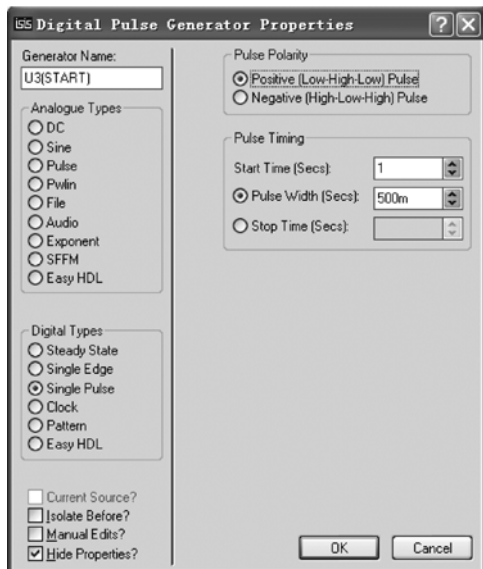


图 4-7 AD 转换启动信号

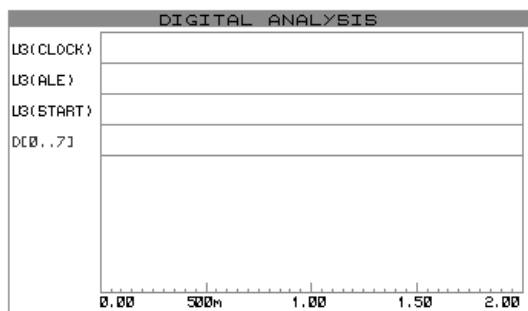


图 4-8 在数字仿真图表中添加信号源

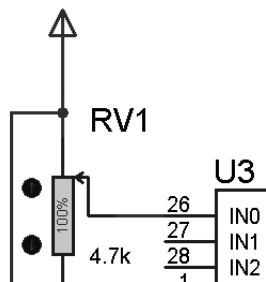


图 4-9 滑动变阻器为 100%

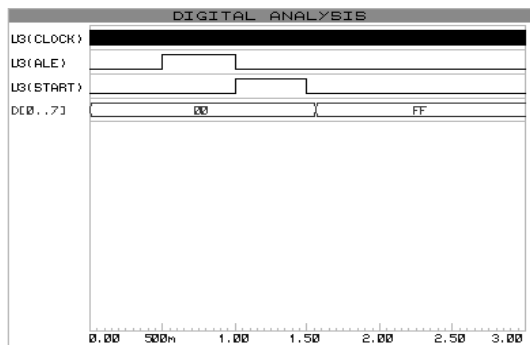


图 4-10 仿真电路（滑动变阻器设置为 100%）

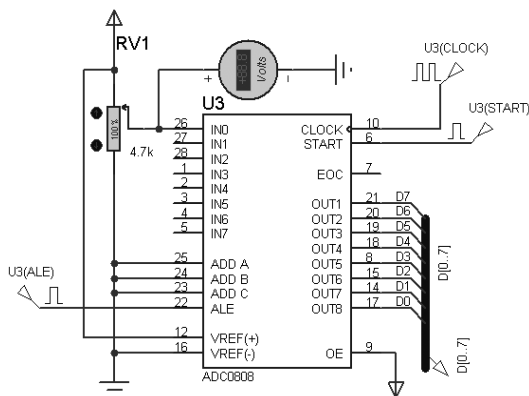


图 4-11 在电路的模拟信号输入端口连接电压表

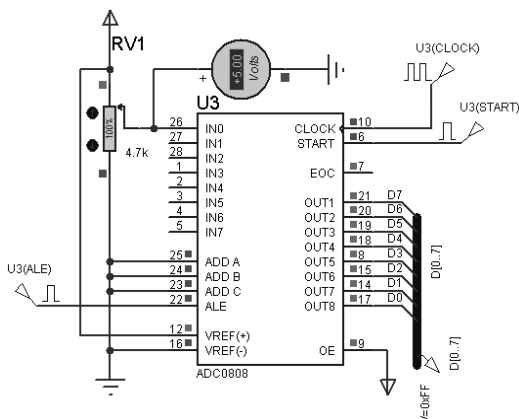


图 4-12 电路仿真结果（测量输入电压）

仿真电路。将滑动变阻器的值设置为 50%，如图 4-13 所示。

单击控制面板中的运行按钮，此时电路的仿真结果如图 4-14 所示。从仿真结果可知，此时系统的模拟输入电压为 2.5V。

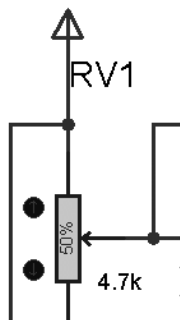


图 4-13 设置滑动变阻器的值为 50%

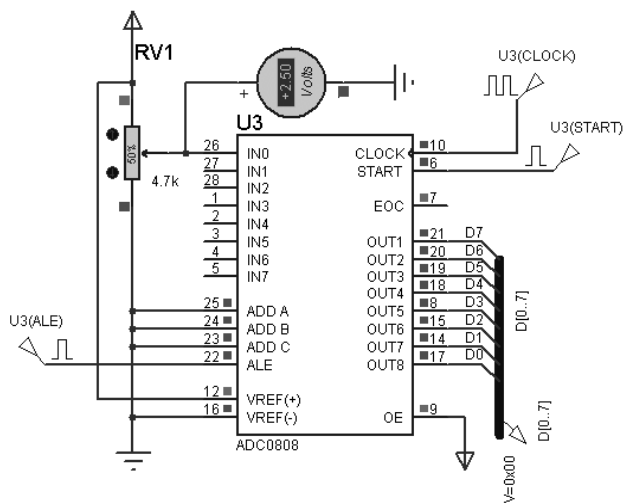


图 4-14 仿真结果（测量滑动变阻器为 50% 时电路模拟输入电压大小）

单击控制面板中的停止按钮，停止仿真。将鼠标放置到数字图表中，按 Space 键，系统数字分析结果如图 4-15 所示。

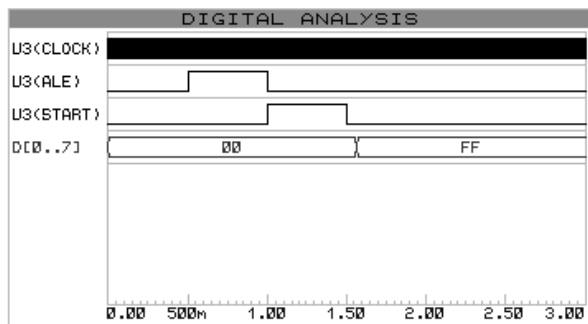


图 4-15 输入模拟电压为 2.5V 时系统的数字输出

从系统的仿真结果可知，这一电路可实现模数转换。

4.1.3 设计 PWM 输出控制电路

【AT89C51 简介】 AT89C51 是一种带 4KB 闪烁可编程可擦除只读存储器（Flash Programmable and Erasable Read Only Memory）的低电压、高性能 CMOS 8 位微处理器。该器件采用 Atmel 高密度非易失存储器制造技术制造，与工业标准的 MCS-51 指令集和输出引脚相兼容。由于将多功能 8 位 CPU 和闪烁存储器组合在单个芯片中，Atmel 的 AT89C51 是一种高效微控制器，为很多嵌入式控制系统提供了一种灵活性高且价廉的方案。其主要特性如下：

- ☉ 与 MCS-51 兼容;
- ☉ 4KB 可编程闪烁存储器;
- ☉ 寿命: 1 000 写/擦循环;
- ☉ 数据保留时间: 10 年;
- ☉ 全静态工作: 0~24Hz;
- ☉ 三级程序存储器锁定;
- ☉ 128×8 位内部 RAM。
- ☉ 32 条可编程 I/O 线;
- ☉ 两个 14 位定时器/计数器;
- ☉ 5 个中断源;
- ☉ 可编程串行通道;
- ☉ 低功耗的闲置和掉电模式;
- ☉ 片内振荡器和时钟电路;

P0 口为一个 8 位漏极开路双向 I/O 口, 每脚可吸收 8TTL 门电流。当 P1 口的引脚第一次写 1 时, 被定义为高阻输入。P0 口能够用于外部程序数据存储器, 它可以被定义为数据/地址的第八位。在 FLASH 编程时, P0 口作为原码输入口, 当 FLASH 进行校验时, P0 口输出原码, 此时 P0 口外部必须被拉高。

P1 口、P2 口及 P3 口内部均为提供上拉电阻的 8 位双向 I/O 口, P1 口缓冲器能接收输出 4TTL 门电流。

使用 AT89C51 的 P2.4、P2.5、P2.6 及 P2.7 引脚分别与 ADC0808 时钟端、AD 转换完成信号端、AD 转换起始端及允许输出端相连, 并且 ADC0808 的输出信号通过 P1 引脚输入到单片机中, 电路图如图 4-16 所示。

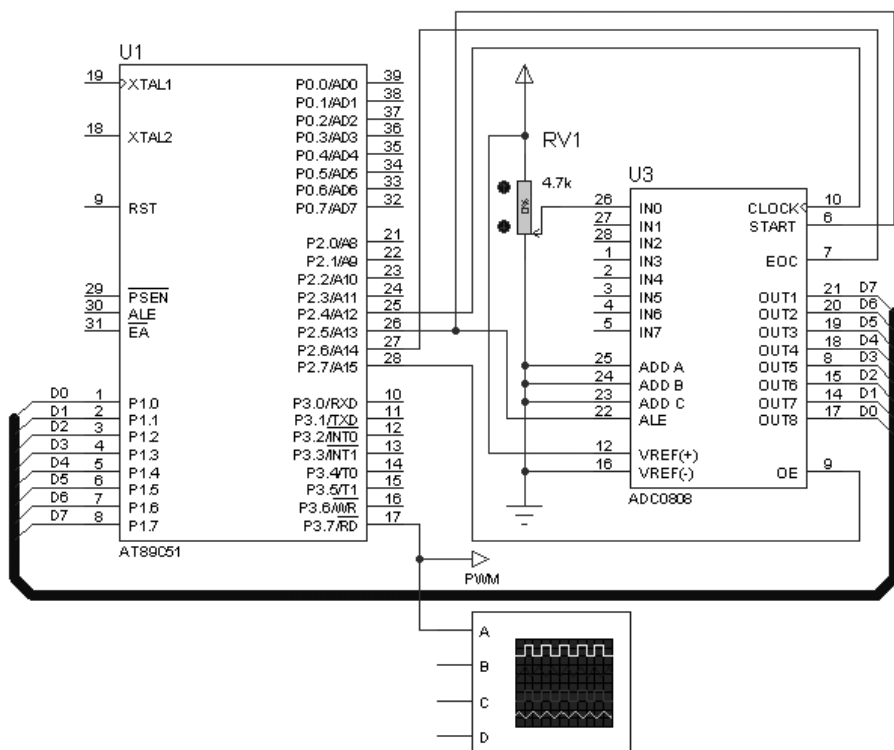


图 4-16 PWM 输出控制电路

其中 AT89C51 的设置如图 4-17 所示。

系统时钟频率为 12MHz。

在制作实际电路时, 需为电路设计晶振电路及复位电路。其中系统的晶振电路如图 4-18

所示，系统复位电路如图 4-19 所示。

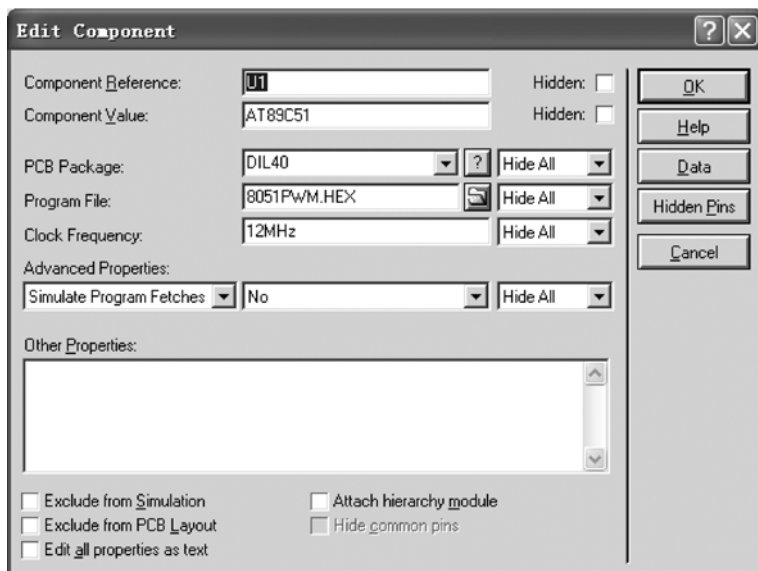


图 4-17 AT89C51 的设置

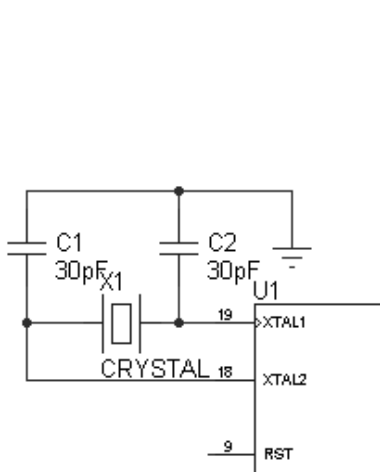


图 4-18 晶振电路

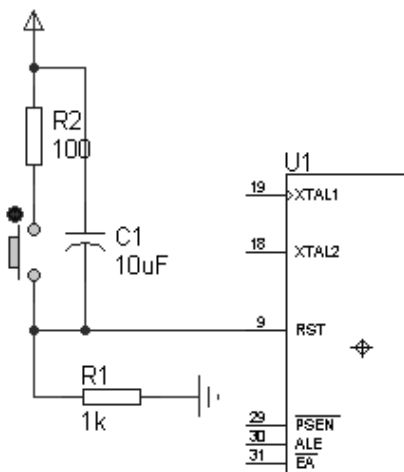


图 4-19 系统复位电路



4.2 在 PROTEUS ISIS 中进行软件编程

【PROTEUS ISIS 的源代码】 PROTEUS VSM 源代码控制系统包含以下两个主要特性：

- ☉ 程序源代码置于 ISIS 中。这一功能使得用户可以直接在 ISIS 编辑环境中直接编辑源代码，而无须手动切换应用环境。
- ☉ 在 ISIS 中定义了源代码编译为目标代码的规则。一旦程序启动，并执行仿真，这些规则将被实时加载，因此目标代码被更新。

说明:

如果用户定义的汇编程序或编译器自带 IDE，则可直接在其中编译，无须使用 ISIS 提供的源代码控制系统。当生成外部程序时，切换回 PROTEUS 即可。

4.2.1 在 PROTEUS ISIS 中创建源代码文件

单击工具栏中的 Source Code 图标，如图 4-20 所示，出现如图 4-21 所示界面。



图 4-20 Source Code 图标

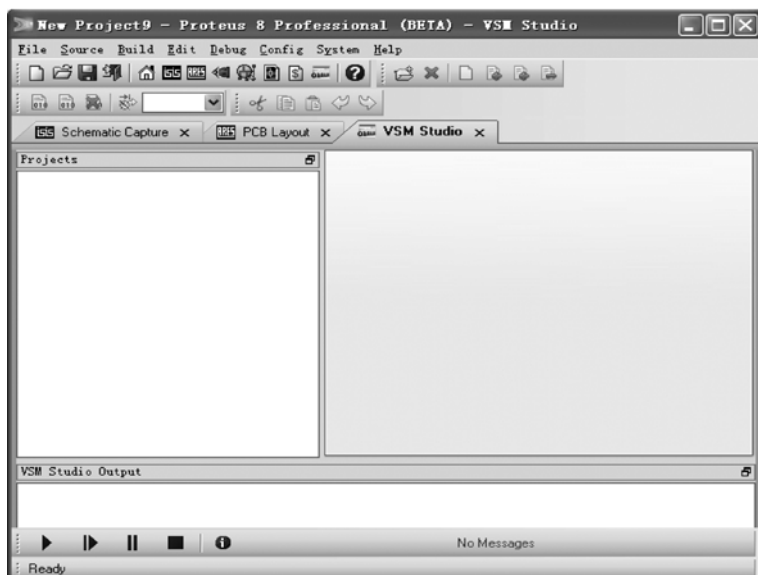


图 4-21 源文件编辑界面

(1) 选择 Source→Create Project 命令，如图 4-22 所示，弹出如图 4-23 所示的代码生成工具列表。

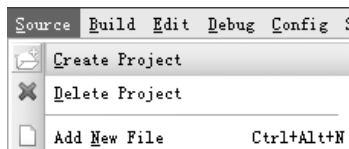


图 4-22 选择 Source→Create Project 命令

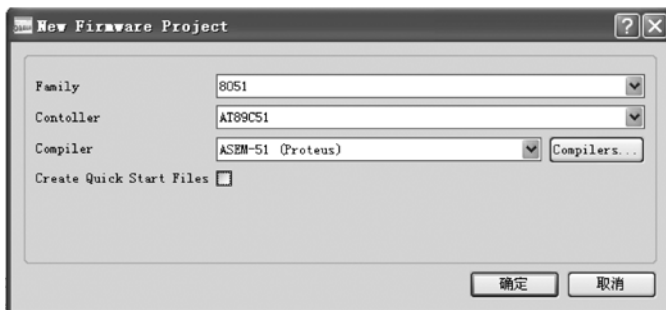


图 4-23 代码生成工具列表

在本例中微处理器为 AT89C51，因此选择 ASEM-51 代码生成工具。若要自己建立新的

源代码, 则取消选中 Create Quick Start Files 选项, 如图 4-23 所示。

(2) 选择 Source→Add New File 命令, 将弹出如图 4-24 所示的 Add New File 对话框。保存为 8051PWM.ASM, 如图 4-25 所示。



图 4-24 Add New File 对话框



图 4-25 源文件编辑窗口

单击“保存”按钮, 这样已经将 8051PWM.ASM 添加到 Source Files 中, 如图 4-26 所示。双击 8051PWM.ASM, 即可打开源文件编辑窗口, 如图 4-27 所示。在编辑环境中输入程序。

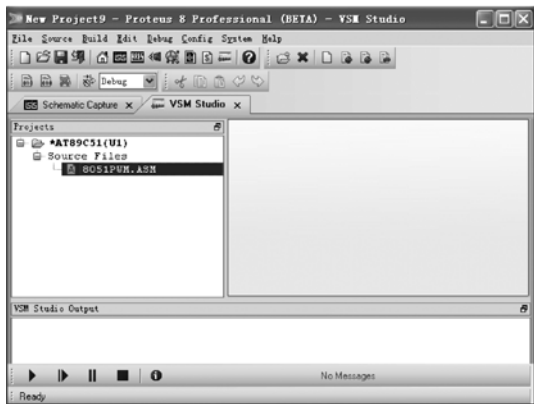


图 4-26 添加 8051PWM.ASM 文件到源文件中

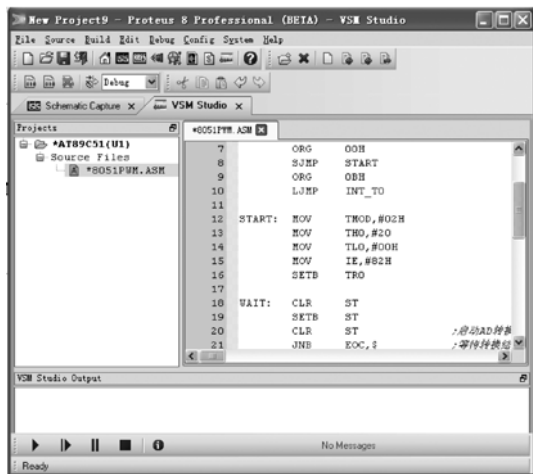


图 4-27 源文件编辑窗口

PWM 输出控制电路软件源程序如下:

ADC	EQU	35H	
CLOCK	BIT	P2.4	; 定义 ADC0808 时钟位
ST	BIT	P2.5	
EOC	BIT	P2.6	
OE	BIT	P2.7	
PWM	BIT	P3.7	
	ORG	00H	
	SJMP	START	


```

        ORG      0BH
        LJMP     INT_T0

START:   MOV      TMOD,#02H
        MOV      TH0,#20
        MOV      TL0,#00H
        MOV      IE,#82H
        SETB     TR0

WAIT:    CLR      ST
        SETB     ST
        CLR      ST                ; 启动 AD 转换
        JNB      EOC,$            ; 等待转换结束
        SETB     OE
        MOV      ADC,P1          ; 读取 AD 转换结果
        CLR      OE
        SETB     PWM            ; PWM 输出
        MOV      A,ADC
        LCALL    DELAY
        CLR      PWM
        MOV      A,#255
        SUBB     A,ADC
        LCALL    DELAY
        SJMP     WAIT

INT_T0:  CPL      CLOCK          ; 提供 ADC0808 时钟信号
        RETI

DELAY:   MOV      R5,#1
D1:      DJNZ     R5,D1
        DJNZ     ACC,D1
        RET


        END
```

编辑完成后，选择 File→Save Project 命令，保存源文件。

4.2.2 在 PROTEUS ISIS 中将源代码文件生成目标代码

在源程序编辑窗口，选择 Build→Build Project 命令，如图 4-28 所示。执行这一命令后，ISIS 将会运行相应的代码生成工具，对所有源文件进行编译、链接，生成目标代码，同时在 VSM Studio Output 显示相关内容，双击 VSM Studio Output，以窗口形式出现，如图 4-29 所示。

这一创建信息给出了关于源代码的编译信息。本例中的源代码没有语法错误，并且 PROTEUS ISIS 将源代码生成了目标代码。


注意：

8.0 版本默认的 HEX 文件的名称是“Debug”，默认的路径和文件名在一个文件夹里。不同的文件会自动建立不同的文件夹。

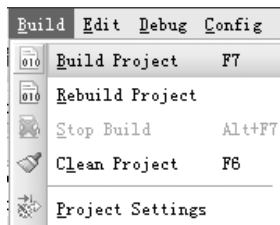


图 4-28 选择 Build→Build Project 命令

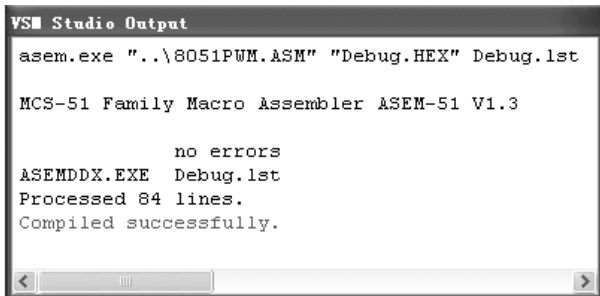


图 4-29 VSM Studio Output 窗口



4.3 进行系统调试

【PROTEUS ISIS 中的源代码调试】PROTEUS VSM 支持源代码调试。系统的 debug loaders 包含在系统文件 LOADERS.DLL 中。目前，系统可支持的工具的数量正在迅速增加。

对于系统支持的汇编程序或编译器，PROTEUS VSM 将会为设计项目中的每一个源代码文件创建一个源代码窗口，并且这些代码将会在 Debug 菜单中显示。

在进行代码调试时，须先在微处理器属性编辑中的 Program File 项配置目标代码文件名（通常为 HEX、S19 或符号调试数据文件（symbolic debug data file））。ISIS 不能自动获取目标代码，因为，在设计中可能有多个处理器。

4.3.1 将目标代码添加到电路

在 PROTEUS ISIS 编辑环境中，双击 AT89C51，将弹出如图 4-30 所示的 AT89C51 元件属性编辑对话框。

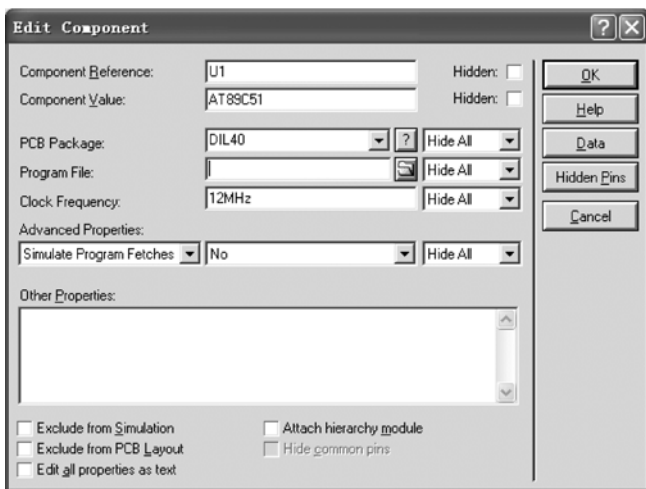


图 4-30 AT89C51 元件属性编辑对话框

单击 Program File 文本框中的打开按钮，如图 4-31 所示，将弹出如图 4-32 所示的文件浏览窗口。



图 4-31 Program File 文本框中的打开按钮

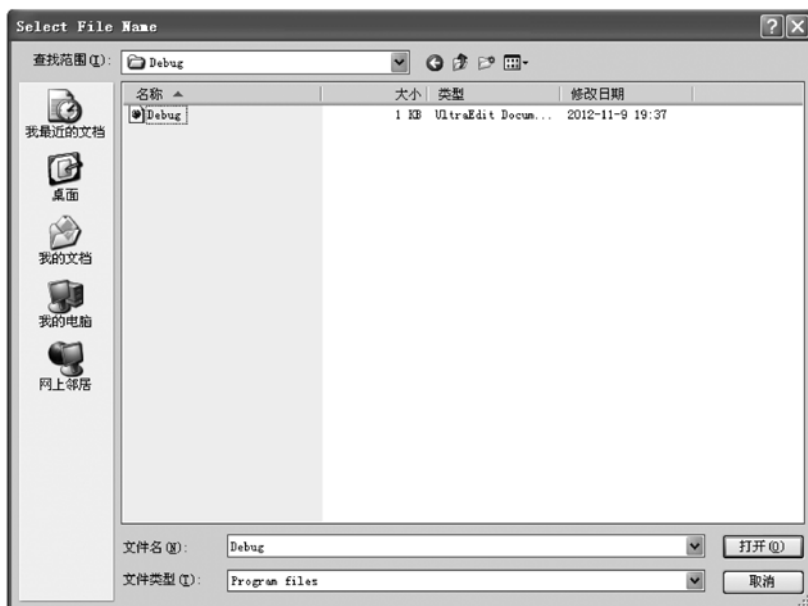


图 4-32 文件浏览窗口

选择 8051PWM.HEX 文件后，单击“打开”按钮，此时就将目标代码添加到了电路中，如图 4-33 所示。

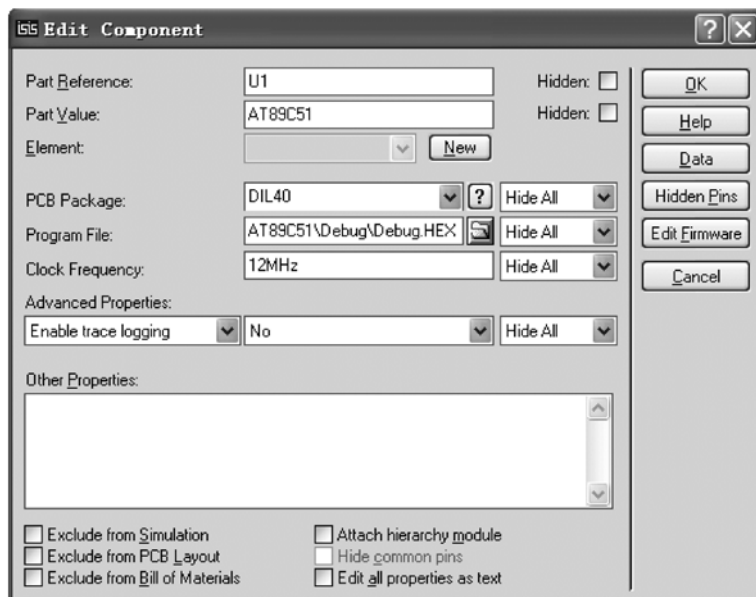



图 4-33 添加目标代码到电路

单击“OK”按钮完成编辑。

4.3.2 进行电路调试

单击控制面板中的暂停按钮, 开始调试程序。此时系统弹出源代码窗口, 如图 4-34 所示。

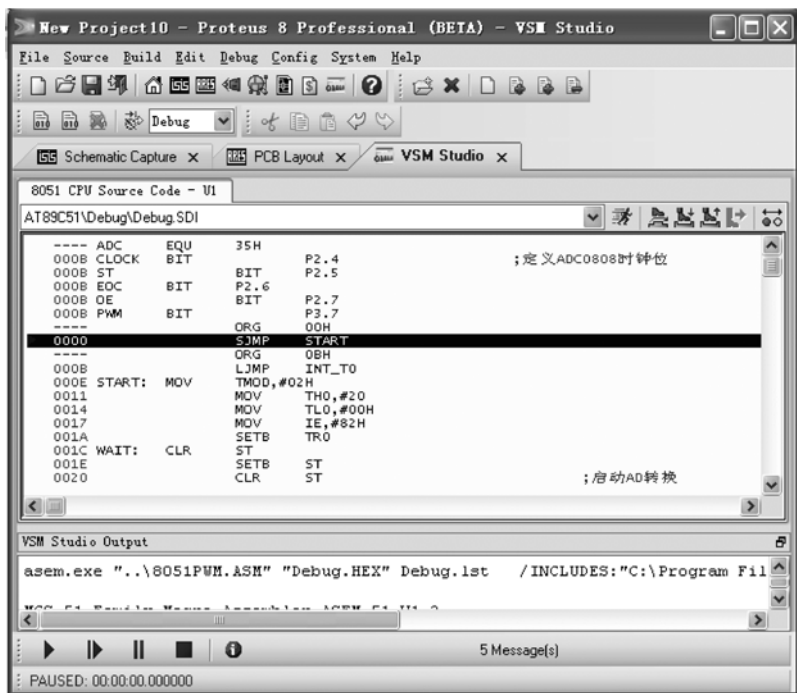






图 4-34 源代码窗口

源代码窗口具有以下特性:

- ☺ 源代码窗口为一组合框, 允许用户选择组成项目的其他源代码文件。用户也可使用快捷键“Ctrl+1”、“Ctrl+2”、“Ctrl+3”等切换源代码文件。
- ☺ 蓝色的条代表当前命令行, 在此处按 F9 键, 可设置断点; 如果按 F10 键程序将单步执行。
- ☺ 前面的红色箭头表示处理器程序计数器的当前位置。
- ☺ 红色圆圈标注的行说明系统在这里设置了断点。

在源代码窗口系统提供了如下命令按钮:

-  Step Over: 执行下一条指令。在执行到子程序调用语句时, 整个子程序将被执行。
-  Step Into: 执行下一条源代码指令。如果源代码窗口未被激活, 系统将执行一条机器代码指令。
-  Step Out: 程序一直执行, 直到当前的子程序返回。
-  Step To: 程序一直在执行, 直到程序到达当前行。这一选项只在源代码窗口被激活的状况下可用。

除 Step To 选项外, 单步执行命令可在源代码窗口不出现的状况下使用。

在源代码窗口右击, 将出现如图 4-35 所示的右键菜单。

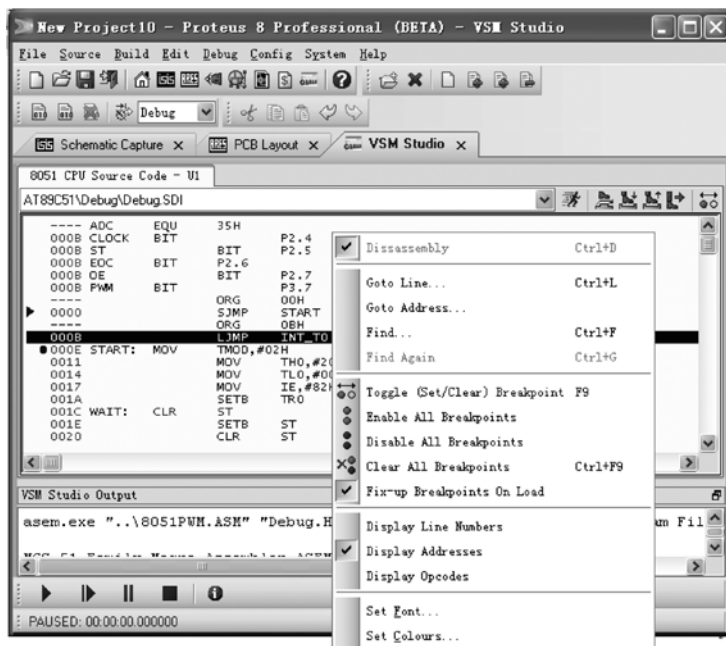


图 4-35 源代码窗口中的右键菜单

右键菜单提供了许多功能选项，其中 Display Line Numbers 为显示行号，如图 4-36 所示。

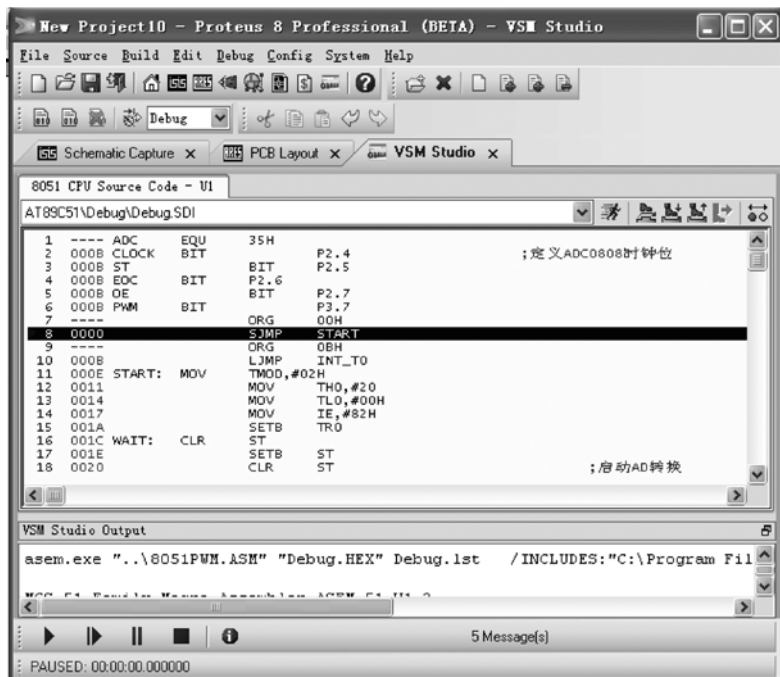


图 4-36 在源代码窗口显示行号

Display Opcodes 为显示操作码，如图 4-37 所示。而 Goto Line 为转到行，点选这一命令后，将弹出如图 4-38 所示的对话框。

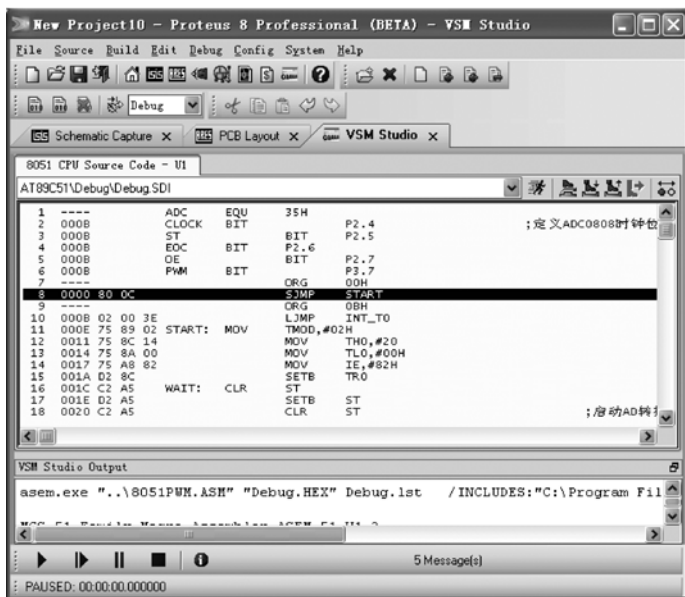


图 4-37 显示操作码

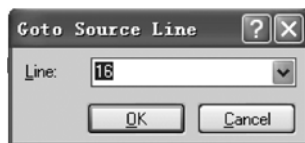


图 4-38 跳转到行对话框

在 Line 文本框中输入待跳转的行号，如 16，单击“OK”按钮，程序中的当前命令行将显示为第 16 行，如图 4-39 所示。

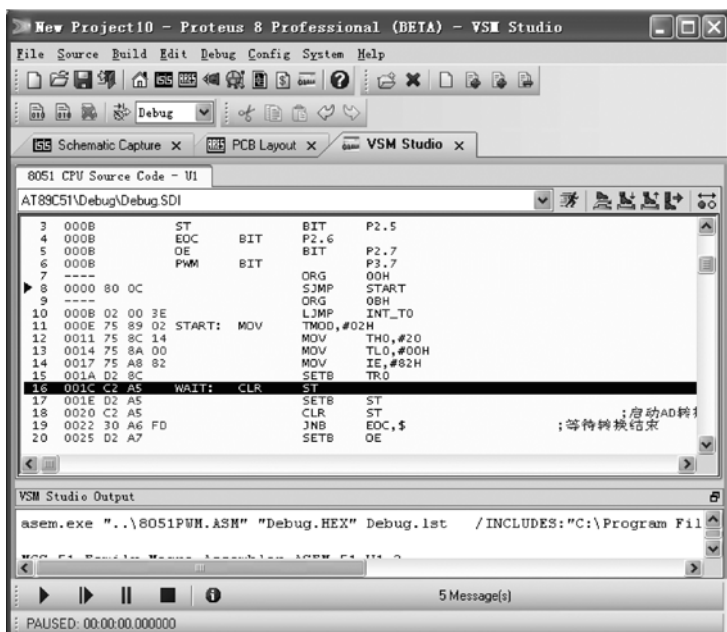



图 4-39 当前命令行为第 16 行

另外，Goto Address 为转到地址，Find Text 为查找文本，Display Addresses 为显示地址等。

当调试高级语言时，用户也可以在显示源代码行和显示系统可执行实际机器代码的列表间切换。机器代码的显示或隐藏可通过“Ctrl+D”快捷键进行设置。

点选 Step Into，执行下一条源代码指令。

当程序执行到如图 4-40 所示的位置时，此条语句为将定时的高位赋值为 20。查看是否赋值到计数器的初值寄存器。选择 Debug→Watch Window 选项，如图 4-41 所示，此时将弹出观测窗口，如图 4-42 所示。

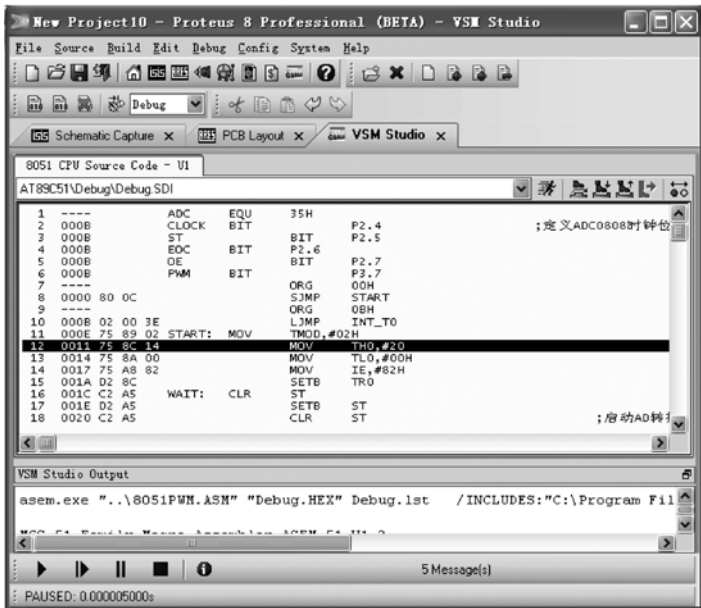


图 4-40 程序执行到赋值语句

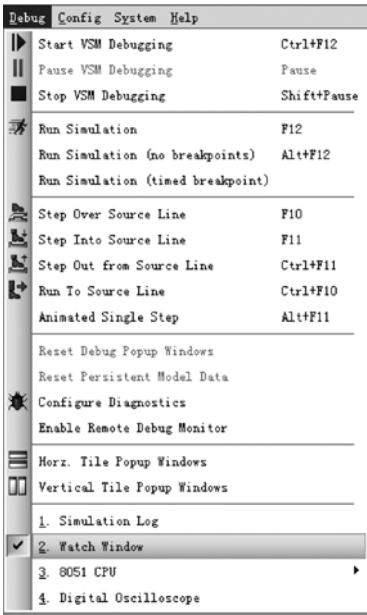


图 4-41 选择 Debug→Watch Window 选项

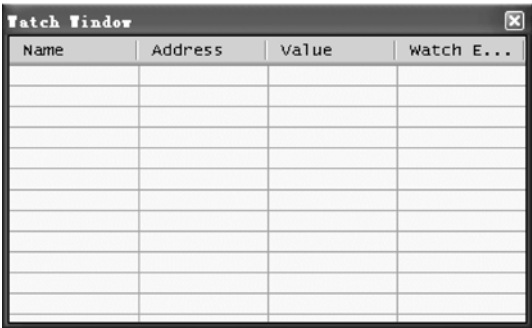


图 4-42 观测窗口

【观测窗口】观测窗口可实时更新显示处理器的变量、存储器的值和寄存器值。它同时还可给独立存储单元指定名称。

在观测窗口中添加项目的步骤如下:

(1) 按“Ctrl+F12”快捷键开始调试,或在系统正处于运行状态时,单击“Pause”按钮,暂停仿真。

(2) 单击 Debug 菜单中的窗口序号,显示 Watch Window 窗口。

(3) 在 Watch Window 窗口右击,将弹出如图 4-43 所示的右键菜单。

其中 Add Items (by Name) 为按名称添加项目, Add Items (by Address) 为按地址添加项目。点选 Add Items (by Name), 将出现如图 4-44 所示的对话框。

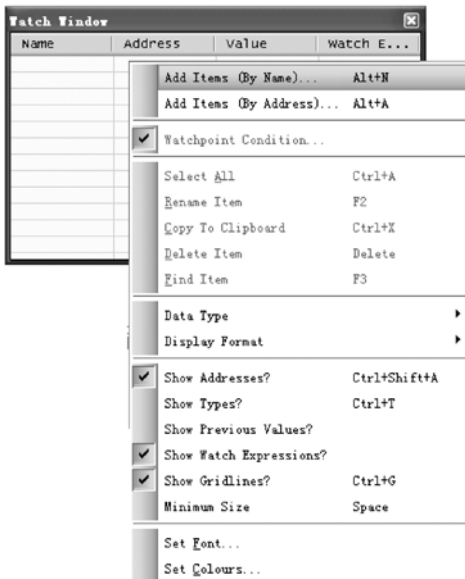


图 4-43 Watch Window 窗口右键菜单

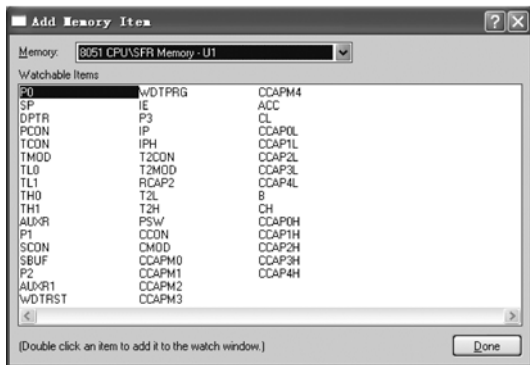


图 4-44 按名称添加项目对话框

如果电路中包含多个 CPU,则可点选 Memory 的下拉式按钮,选择期望的 CPU。

双击希望观测的变量,变量将添加到观测窗口。如双击 SCON 变量,SCON 变量将被添加到观测窗口,如图 4-45 所示。

若使用 Add Items (by Address) 命令添加项目到观测窗口,将出现如图 4-46 所示的对话框。

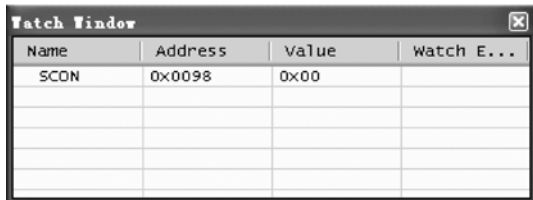


图 4-45 添加 SCON 变量到观测窗口

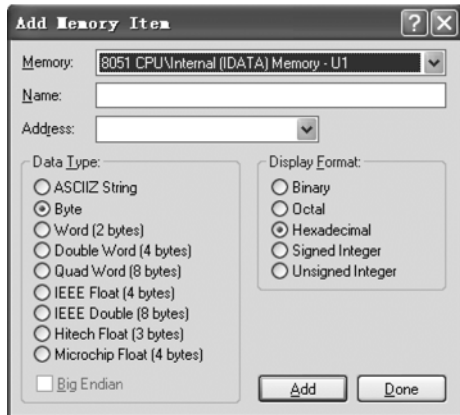


图 4-46 按地址添加项目对话框

点选 Memory 的下拉式按钮，可选择其他寄存器，如图 4-47 所示。

选择期望观测的寄存器后，在 Name 文本框中输入名称，在 Address 文本框中输入地址，即可将项目添加到 Watch Window 窗口。如在 Name 文本框中键入 data1，在 Address 文本框中输入 0x0018，数据类型设置为 Byte（字节），数据显示方式设置为 Hexadecimal，如图 4-48 所示。

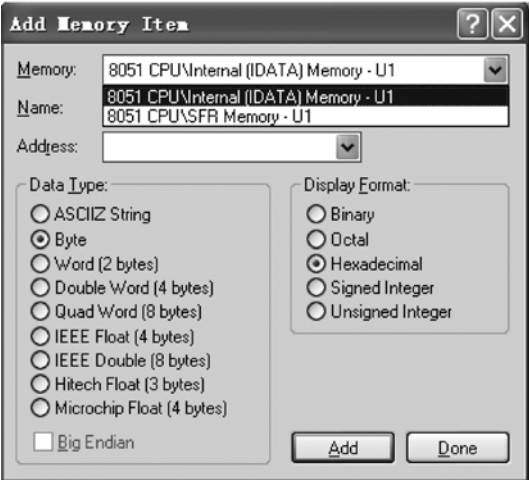


图 4-47 选择其他寄存器

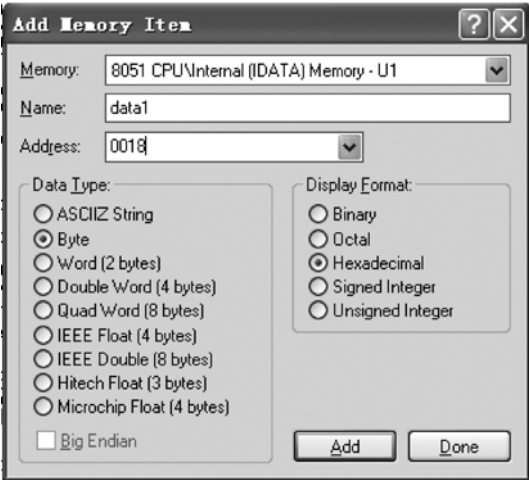


图 4-48 添加 0x0018 后的窗口

单击“Add”按钮，0x0018 地址的数据将被添加到观测窗口，如图 4-49 所示。

当数据格式不便于观测时右击，在弹出的右键菜单中选择 Display Format 命令，系统将列出如图 4-50 所示的数据格式。

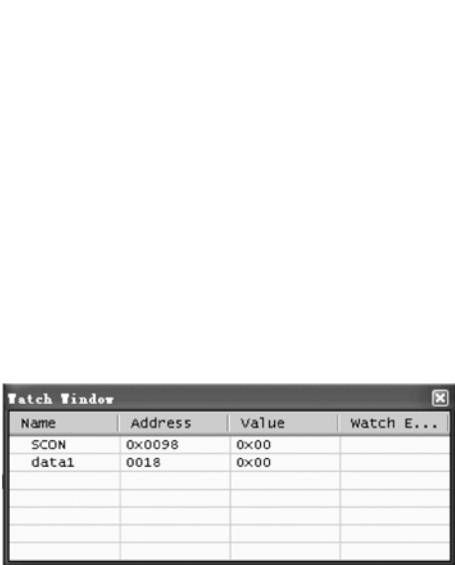


图 4-49 添加 0x0018 地址的数据到观测窗口

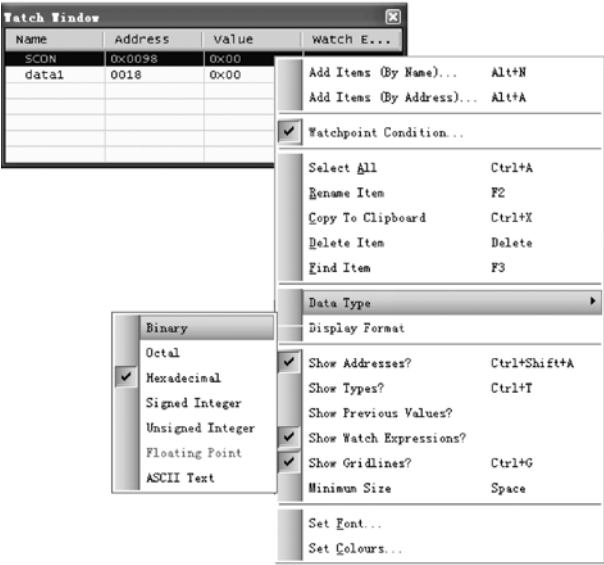


图 4-50 PROTEUS ISIS 提供的的数据格式

系统提供如二进制、八进制、十进制或十六进制等数据形式。点选 Binary（二进制）选项，则观测窗口的数据格式以二进制形式显示，如图 4-51 所示。

在观测窗口可设置观测点。当项目的值与观测点设置条件相符时，观测窗口可延缓仿真。

按“Ctrl+F12”快捷键开始调试，或在系统正处于运行状态时，单击“Pause”按钮，暂停仿真。单击 Debug 菜单中的窗口序号，显示 Watch Window。添加需要观测的项目，點選需要设置观测点的观测项目，并单击窗口的右键菜单中的 Watchpoint Condition 命令，如图 4-52 所示。

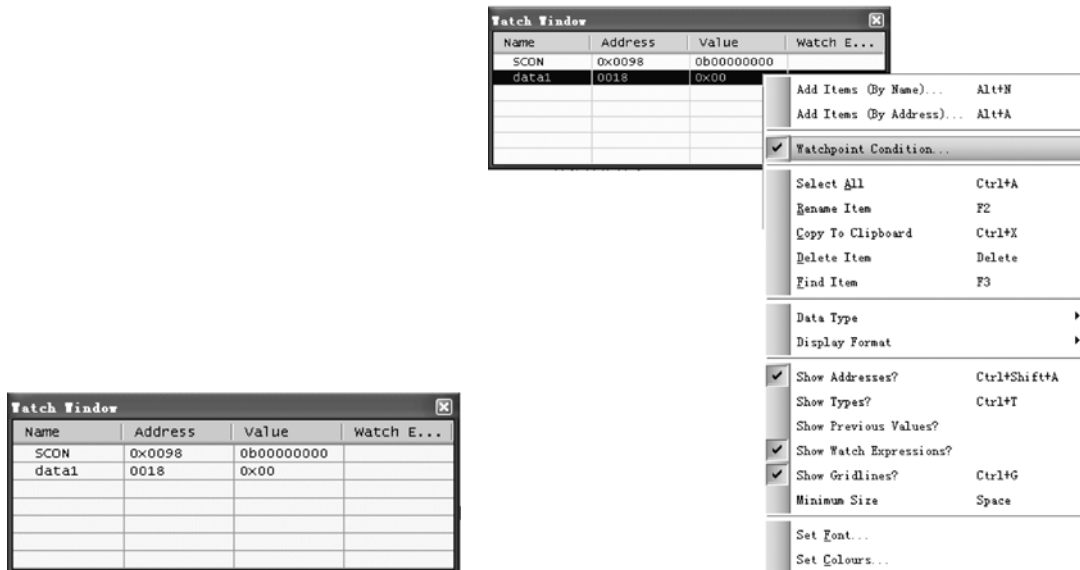


图 4-51 SCON 数据以二进制形式显示

图 4-52 选择观测点的观测条件命令

此时将出现如图 4-53 所示的观测点设置对话框。各选项介绍如下：

☺ Global Break Condition: 设置观测方式。

- ☞ Turn off (disable) watch points: 关闭观测功能。
- ☞ Suspend the simulation if ANY expression is true: 任一表达式为真时，延缓仿真。
- ☞ Stop the simulation only when ALL expressions are true: 所有表达式为真时，停止仿真。

☺ Item Break Expression: 观测点观测表达式。

- ☞ Item: 观测项目。
- ☞ Mask: 屏蔽方式及屏蔽操作数。屏蔽方式包括与、或及异或等，如图 4-54 所示。
- ☞ Condition: 操作算符。所包含的操作算符如图 4-55 所示。
- ☞ Value: 操作数。

按图 4-56 所示设置观测点。

观测设置为：任一表达式为真时，延缓仿真模式；观测项目为 data1，屏蔽方式设置为无，观测操作算符设置为 Equals（相等），而操作数设置为 10。即当 data1=10 时，系统暂停仿真。

设置完成后单击“OK”按钮，即可完成设置，如图 4-57 所示。在观测窗口右击，选择 Add Items (by Name)，在弹出的添加寄存器项目的窗口，點選待添加的项目，如图 4-58 所示。

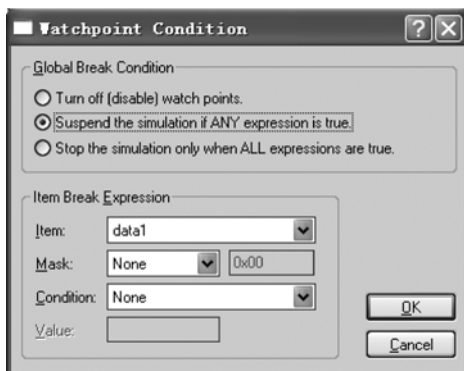


图 4-53 观测点设置对话框

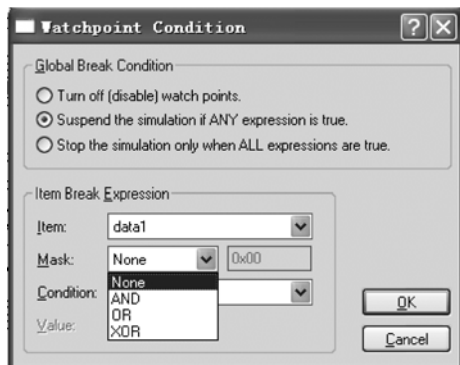


图 4-54 屏蔽方式

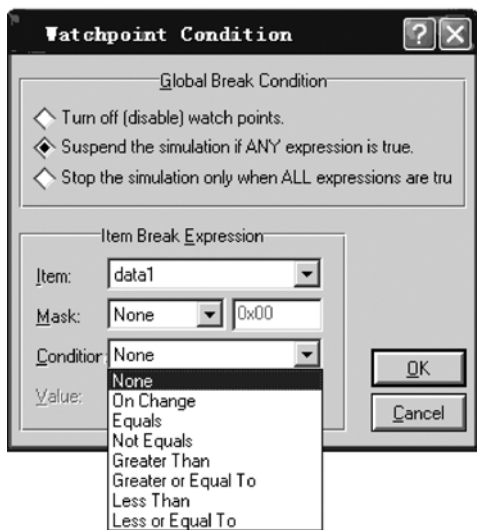


图 4-55 操作算符

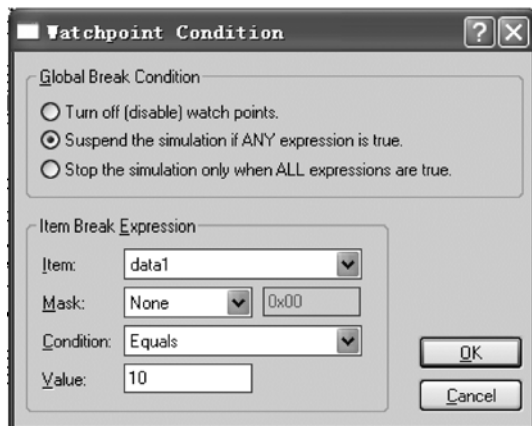


图 4-56 设置观测点

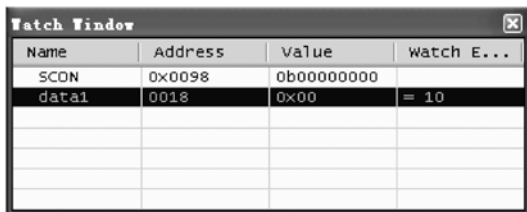


图 4-57 添加观测点观测条件后的 Watch Window 窗口

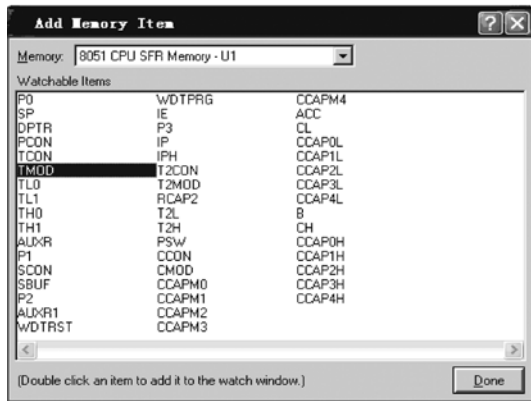



图 4-58 选定项目

双击项目即可添加项目到观测窗口。添加 TMOD、TH0 和 TL1 到编辑窗口，如图 4-59 所示。

点击  Step Into, 执行下一条源代码指令。此时观测窗口各变量值如图 4-60 所示。

Name	Address	Value	Watch E...
TMOD	0x0089	0x02	
TH0	0x008C	0x00	
TL1	0x008B	0x00	

图 4-59 添加 TMOD、TH0 和 TL1 到编辑窗口

Name	Address	Value	Watch E...
TMOD	0x0089	0x02	
TH0	0x008C	0x14	
TL1	0x008B	0x00	

图 4-60 观测窗口各变量值

从观测窗口的数据可知, 观测窗口可实时显示程序执行的结果。



在程序的第 18 行, 单击源代码窗口的设置断点图标 , 即可在第 18 行设置断点, 如图 4-61 所示。





图 4-61 设置断点

单击源代码窗口运行  按钮, 程序会一直执行, 直至运行到断点设置处, 如图 4-62 所示。

将 IE 添加到观测窗口后, 观测窗口数据如图 4-63 所示。

从观测窗口的数据可知, 观测窗口实时显示程序数据。

再次双击源代码窗口的设置断点图标 , 即可取消断点, 如图 4-64 所示。

点击  Step Into, 执行下一条源代码指令。这一代码的意义为清零 P2.5 端口。选择 Debug→8051CPU→Registers-U1 命令, 如图 4-65 所示。

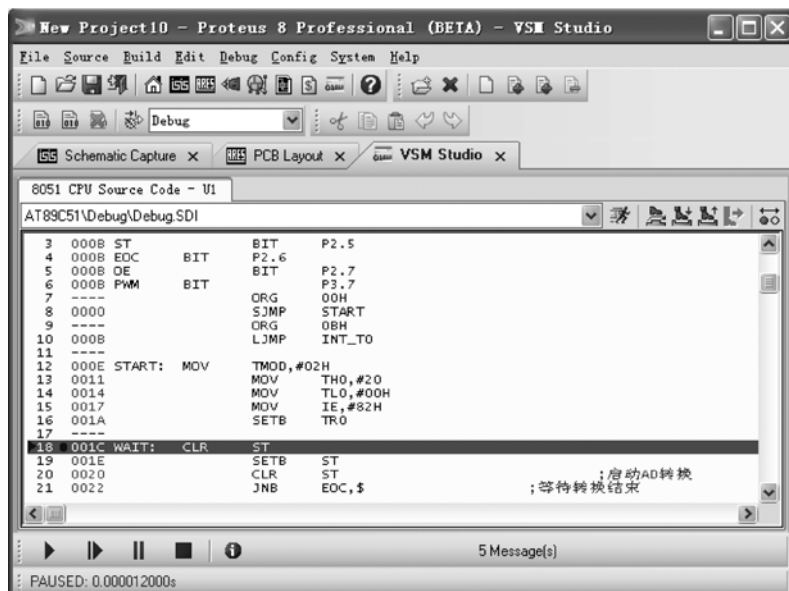


图 4-62 程序运行到断点

Name	Address	Value	Watch E...
TMOD	0x0089	0x02	
TH0	0x008C	0x14	
TL1	0x008B	0x00	
IE	0x00A8	0x82	

图 4-63 程序运行到断点后观测窗口数据

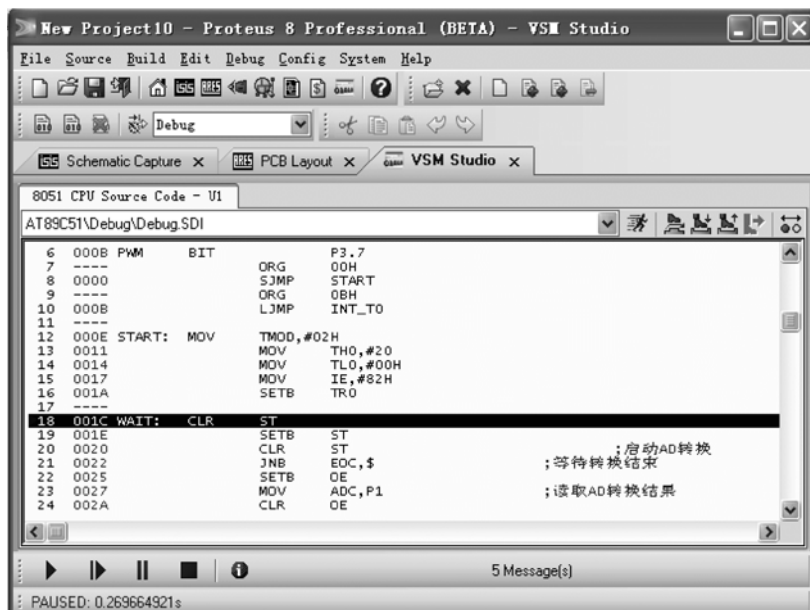


图 4-64 取消断点

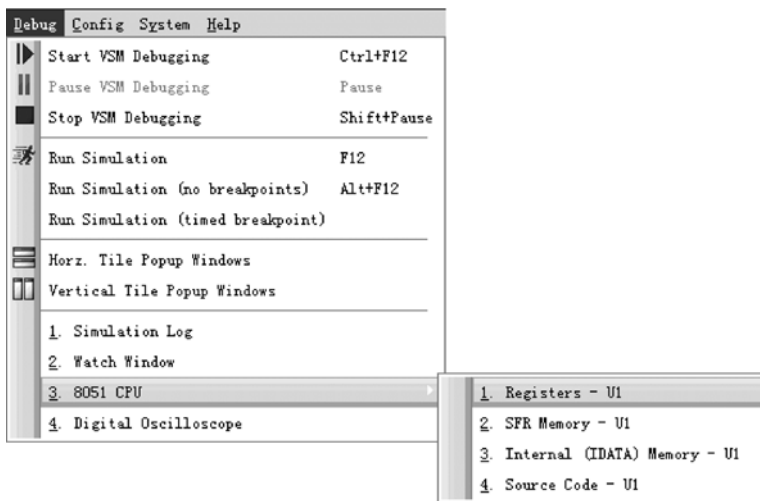


图 4-65 选择 Debug→8051CPU→Registers-U1

此时，将弹出寄存器窗口，双击寄存器窗口标题，将以独立的窗口出现，如图 4-66 所示。再次点选 Step Into，执行下一条源代码指令。这一代码的意义为置位 P2.5 端口，如图 4-67 所示。



图 4-66 清零 P2.5 端口

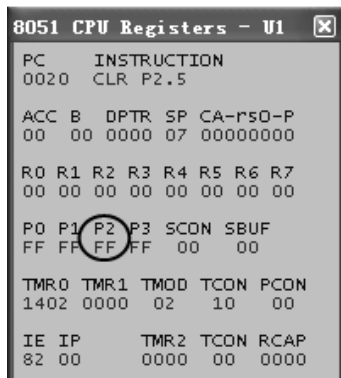


图 4-67 置位 P2.5 端口

第三次点选 Step Into，这一操作将清零 P2.5 端口。这一系列操作用于产生启动 AD 转换脉冲信号。

单击控制面板中的停止按钮，停止仿真。在 START 引脚放置电压探针，如图 4-68 所示。

单击工具箱中的 Simulation Graph 图标，在对象选择器中将出现各种仿真分析所需的图表，选择 INTERACTIVE 交互式仿真图表，如图 4-69 所示。

将探针添加到图表中，探针信号按数字信号处理，如图 4-70 所示。

设置完成后单击“OK”按钮完成设置。

双击交互式仿真图表，设置仿真起始时间为 10μs，停止时间为 25μs，如图 4-71 所示。

单击控制面板中的暂停按钮，打开源代码编辑窗口，在源代码的第 21 行设置断点，如图 4-72 所示。

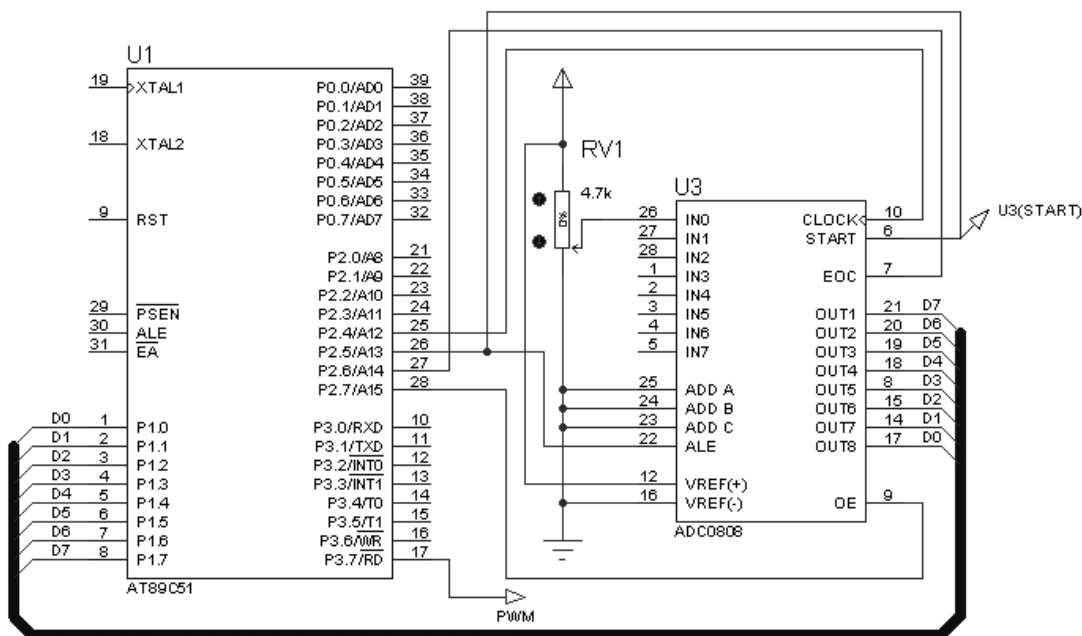


图 4-68 在 START 引脚放置电压探针

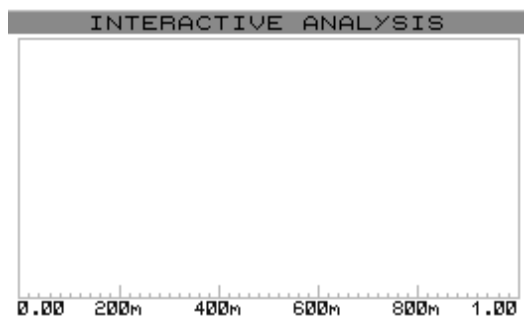


图 4-69 放置交互式仿真图表

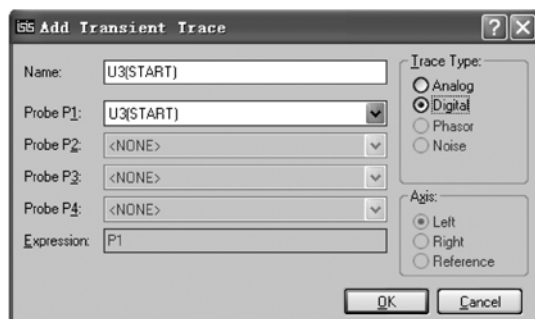


图 4-70 探针信号按数字信号处理添加到图表

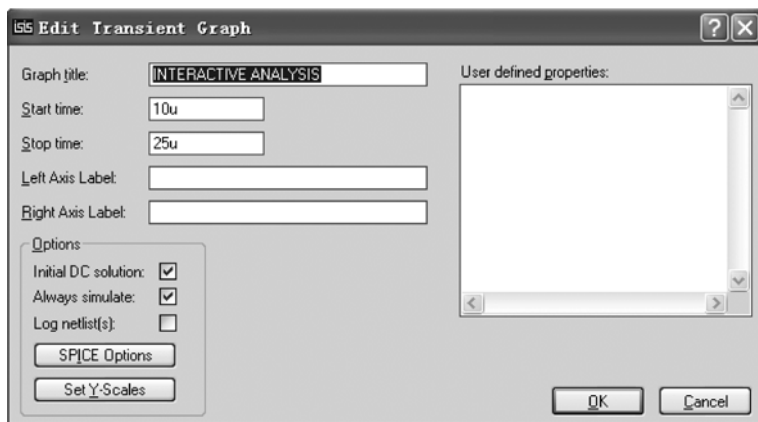


图 4-71 交互式仿真图表属性对话框

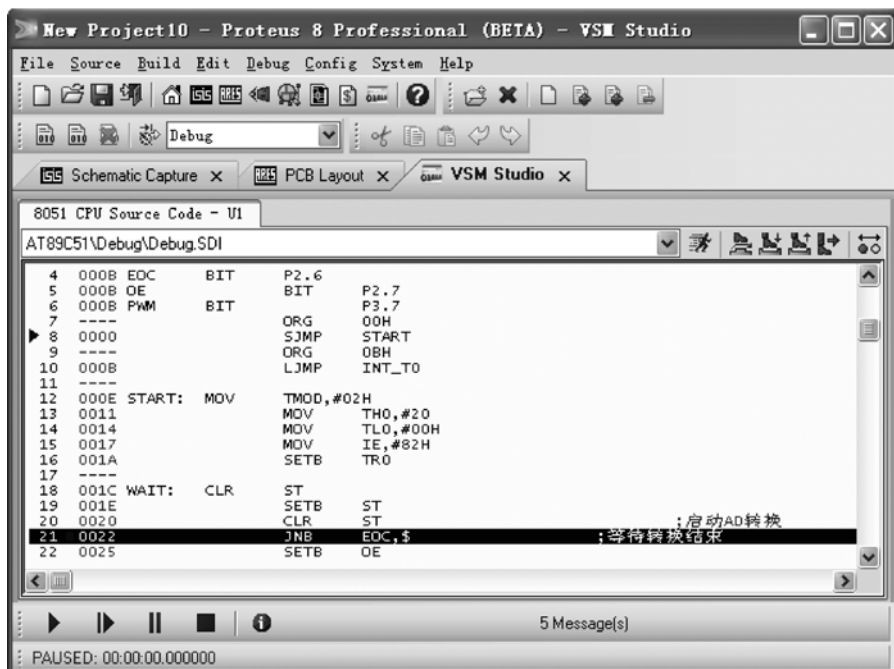


图 4-72 在源代码窗口的第 21 行设置断点

单击控制面板中的停止按钮，停止仿真。然后将鼠标放置到交互式仿真图表中，按空格键仿真电路。电路将在断点处暂停仿真，单击控制面板中的停止按钮，停止仿真，此时交互式仿真图表绘制出 START 端口信号，如图 4-73 所示。

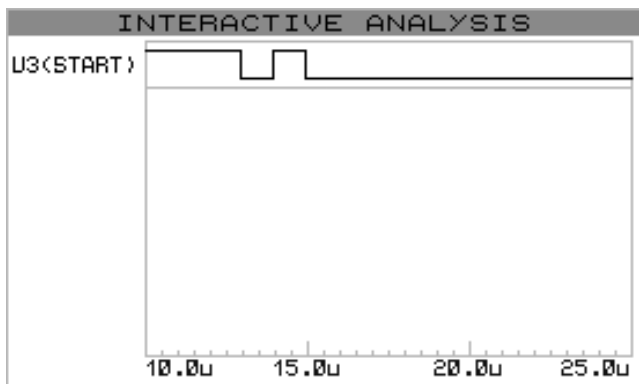


图 4-73 交互式图表绘制出 START 端口信号

按上述方式调试电路，直至程序达到期望的结果。

4.3.3 仿真电路

将 AT89C51 的 PWM 输出端口与示波器的 A 端口相连，如图 4-74 所示。

单击控制面板中的运行按钮，则示波器显示电路输出波形，如图 4-75 所示。

如果不小心或因为其他原因关闭了示波器的输出窗口，可以在仿真运行时选择 Debug→Digital Oscilloscope 命令打开示波器的输出窗口，如图 4-76 所示。

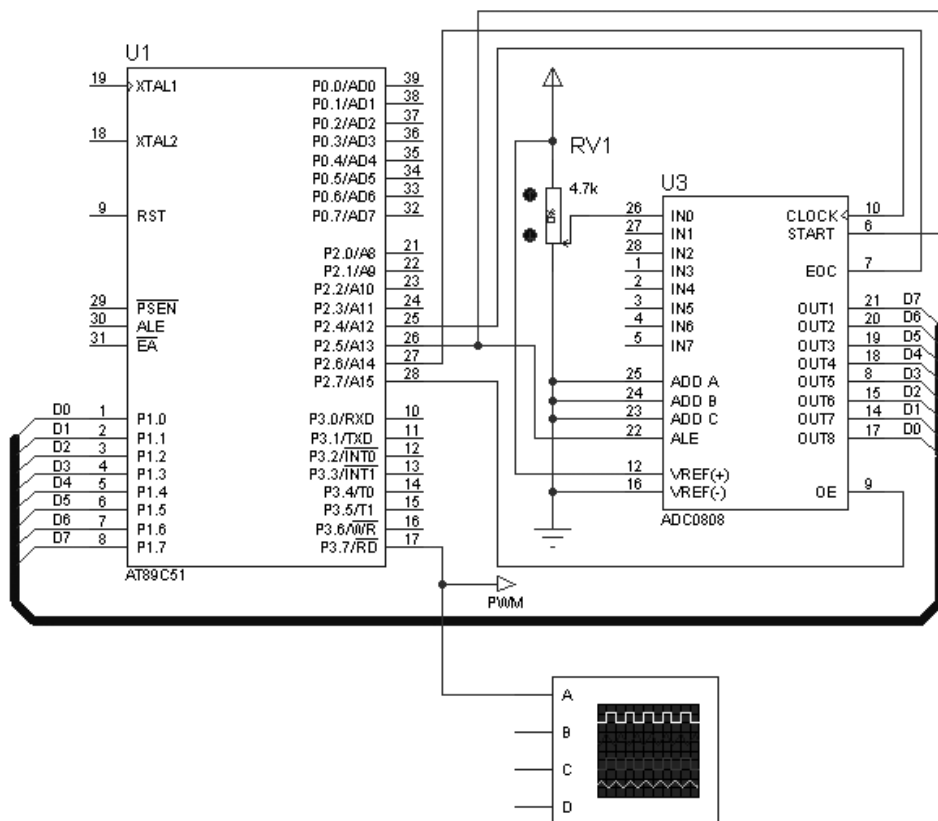


图 4-74 连接 PWM 输出端口与示波器的 A 端口

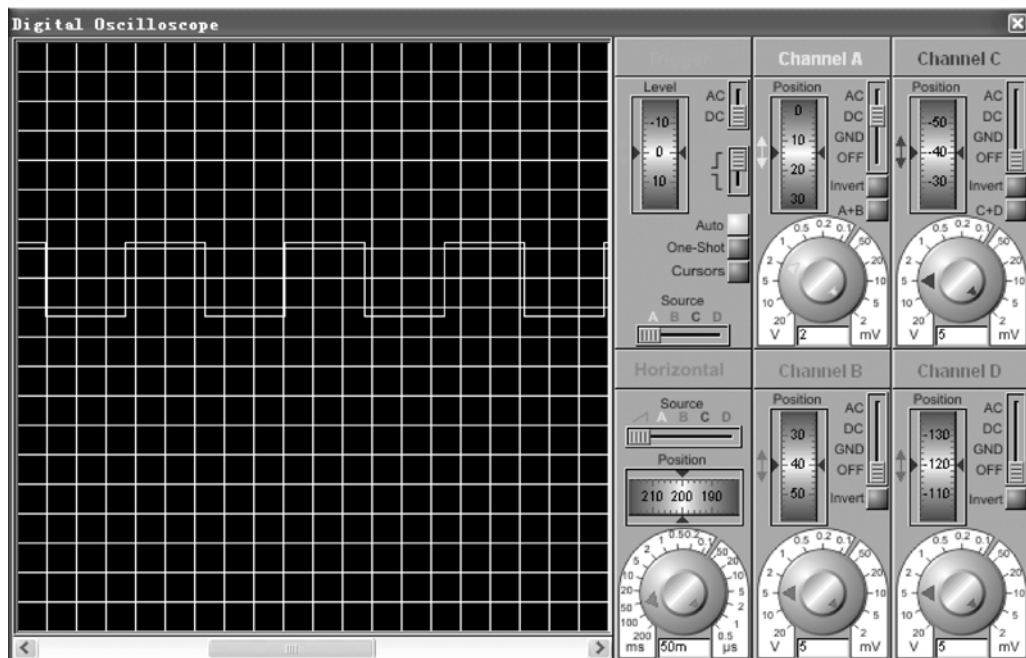


图 4-75 示波器输出结果

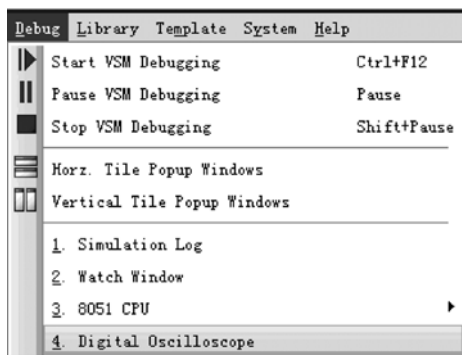


图 4-76 打开示波器输出窗口

放置数字分析图表，测量输出 PWM 波的占空比。将数字分析图表放置到电路编辑窗口，并添加 PWM 变量（在 PWM 波输出端口放置电压探针），如图 4-77 所示。

将鼠标放置到图表中，按 Space 键仿真电路，结果如图 4-78 所示。

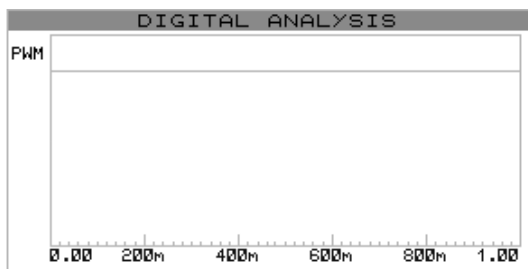


图 4-77 采用数字分析图表测量输出 PWM 波的占空比

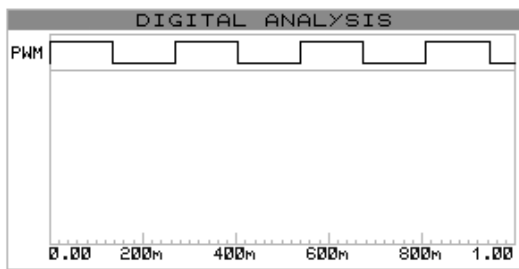


图 4-78 PWM 波数字分析图表仿真结果

单击图表的表头，图表将以窗口形式出现，在窗口中放置测量指针，测量 PWM 波的周期，测量结果如图 4-79 所示。

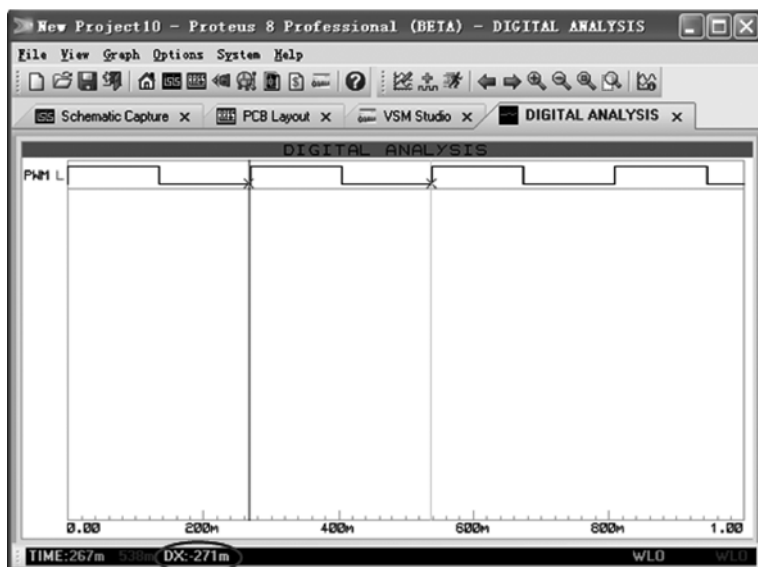


图 4-79 测量 PWM 波周期

从图中的测量结果可知, 系统输出的 PWM 波周期为 0.27s。

按照上述方式测量上升沿的脉宽, 结果如图 4-80 所示。

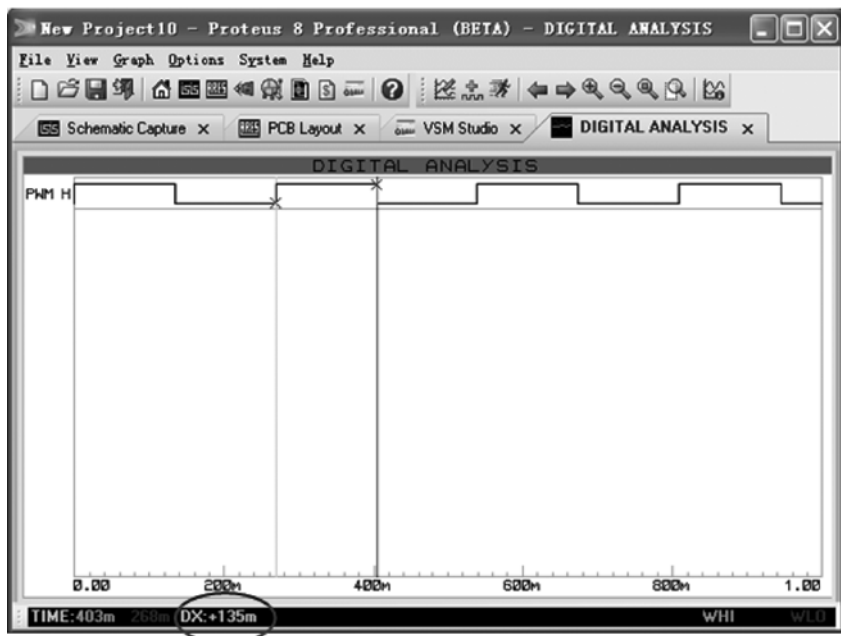


图 4-80 测量 PWM 波上升沿的脉宽

从图中的测量结果可知, 系统输出的 PWM 波上升沿的脉宽为 0.133s。因此此时系统输出的 PWM 波占空比为 1:1。

改变滑动变阻器的阻值, 将其设置为阻值的 20%, 再次仿真电路, 电路输出的波形如图 4-81 所示。

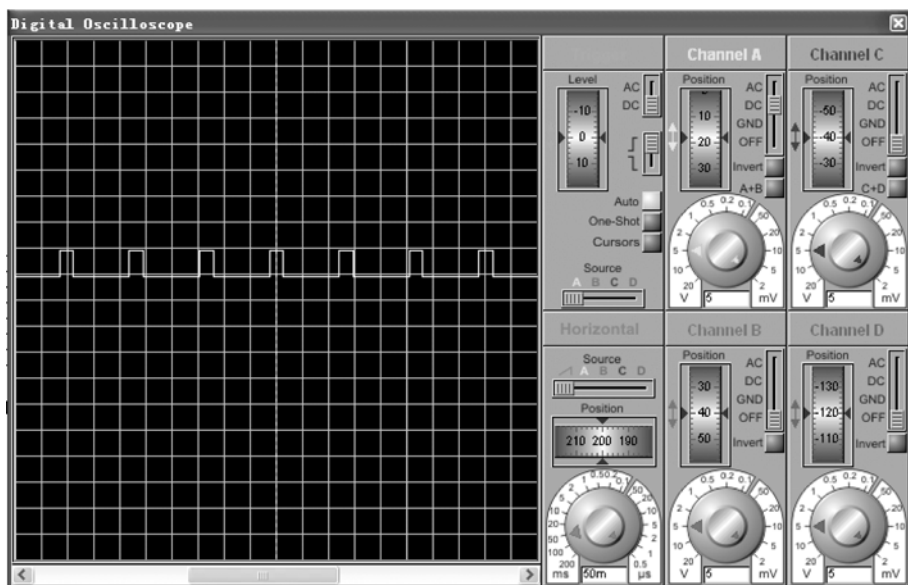


图 4-81 PWM 波 (滑动变阻器阻值设置为 20%时)

采用数字分析图表测量输出 PWM 波的占空比。此时 PWM 波的周期测量结果如图 4-82 所示。从图中的测量结果可知，系统输出的 PWM 波周期为 0.135s。

按照上述方式测量上升沿的脉宽，结果如图 4-83 所示。

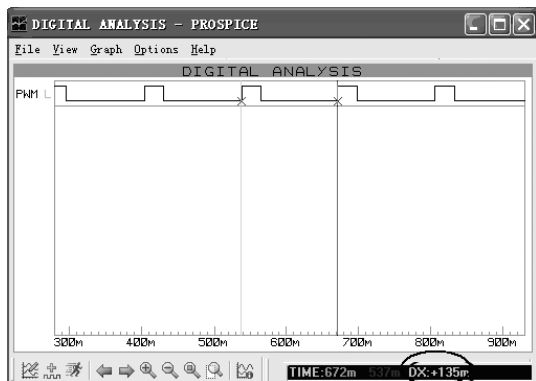


图 4-82 PWM 波周期测量（滑动变阻器阻值为 20%）

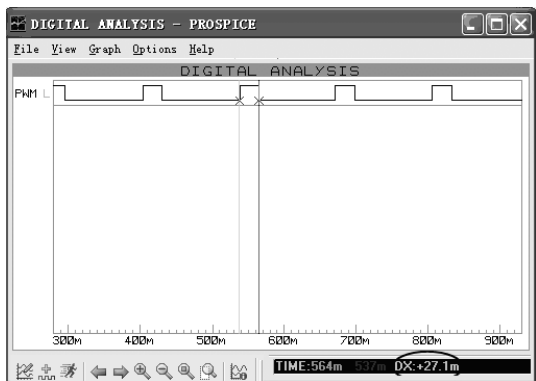


图 4-83 PWM 波上升沿脉宽测量（滑动变阻器阻值为 20%）

从图中的测量结果可知，系统输出的 PWM 波上升沿的脉宽为 0.027 1s。因此此时系统输出的 PWM 波占空比为 1:4。

自此系统调试结束。电路进入电路板制作及实物焊接流程。



4.4 将 PROTEUS 与 Keil 联调

单片机教学包括理论与实践教学，而实践实训教学所占比例较大，硬件投入大。在实践实训中需要大量的实验仪器和设备。一般的学校或个人没有较多的经费。单片机的课堂教学及实验中存在诸多问题，如：

- ⑤ 单片机课堂教学以往多以理论教学为主，实验教学也多是进行验证实验。但单片机是一门实践性很强的课程。教学中需要很多硬件设备，如电脑、仿真机、实验电路、编程器等。一般理论课堂难以辅助硬件进行教学，即便演示，效果也不好，一般单片机实验箱也只是起验证实验的作用。
- ⑥ 学生实验时也有着不少问题，单片机实验室由于存在着场地和时间等问题，学生除了上课外，平时难得有机会实践。个人配备单片机实验开发系统，因成本较高，很多学生无法承受。同时一般单片机实验箱由于是成品，学生很难参与到其中的细节设计中去，学生动手能力很难得到训练与提高。
- ⑦ 实验设备不足、落后，单片机实验室建立成本高，一般学校很少有学生人手一套实验开发系统进行单片机实验及开发。就算有，由于技术不断更新，设备不断老化，实验仪器也会很快落后。要解决此问题需要不断地重建单片机实验室，必会带来资金耗费严重等问题。

PROTEUS 是一种低投资的电子设计自动化软件，提供 Schematic Drawing、SPICE 仿真与 PCB 设计功能，这一点 PROTEUS 与 multisim 比较类似，只不过它可以仿真单片机和周

边设备,可以仿真 51 系列、AVR、PIC 等常用的 MCU。与 Keil 和 MPLAB 不同的是,它还提供了周边设备的仿真,只要给出电路图就可以仿真,如 373、LED、示波器等。PROTEUS 提供了大量的元件库,有 RAM、ROM、键盘、马达、LED、LCD、AD/DA、部分 SPI 器件、部分 IIC 器件,编译方面支持 Keil 和 MPLAB,里面有大量的例子参考。

- ⑤ PROTEUS 软件可仿真数字和模拟、交流和直流等数千种元件和多达 30 多个元件库。
- ⑤ 虚拟仪器仪表的数量、类型和质量是衡量仿真软件实验室是否合格的一个关键因素。在 PROTEUS 软件中,理论上同一种仪器可以在一个电路中随意调用。
- ⑤ 除了现实存在的仪器外,PROTEUS 还提供了一个图形显示功能,可以将线路上变化的信号以图形的方式实时地显示出来,其作用与示波器相似但功能更多。
- ⑤ 这些虚拟仪器仪表具有理想的参数指标,例如极高的输入阻抗、极低的输出阻抗。这些都尽可能减小了仪器对测量结果的影响。
- ⑤ PROTEUS 提供了比较丰富的测试信号用于电路的测试。这些测试信号包括模拟信号和数字信号。

Keil 是德国开发的一个 51 单片机开发软件平台,最开始只是一个支持 C 语言和汇编语言的编译器软件。后来随着开发人员的不断努力以及版本的不断升级,使它已经成为了一个重要的单片机开发平台,不过 Keil 的界面并不是非常复杂,操作也不是非常困难,很多工程师开发的优秀程序都是在 Keil 的平台上编写出来的。可以说它是一个比较重要的软件,熟悉它的人很多很多,用户群极为庞大,操作中有不懂的地方只要找相关的书看看,到相关的单片机技术论坛问问,很快就可以掌握它的基本使用了。

- ⑤ Keil 的 μ Vision3 可以进行纯粹的软件仿真(仿真软件程序,不接硬件电路);也可以利用硬件仿真器,搭接上单片机硬件系统,在仿真器中载入项目程序后进行实时仿真;还可以使用 μ Vision3 的内嵌模块 Keil Monitor-51,在不需要额外的硬件仿真器的条件下,搭接单片机硬件系统对项目程序进行实时仿真。
- ⑤ μ Vision3 调试器具备所有常规源极调试、符号调试特性以及历史跟踪、代码覆盖、复杂断点等功能。DDE 界面和 Shift 语言支持自动程序测试。

为此,利用 PROTEUS 与 Keil 联调,为解决这一问题提供了一些思路。

4.4.1 使用 Keil 的 μ Vision3 集成开发环境

μ Vision3 IDE 是一个 32 位标准的 Windows 应用程序,支持长文件名操作,其界面类似于 MS Visual C++,可以在 Windows 95/98/2000/XP 平台上运行,功能十分强大。 μ Vision3 中包含了一个高效的源程序编辑器、一个项目管理器和一个源程序调试器(MAKE 工具)。

μ Vision3 支持所有的 Keil8051 工具,包括 C 编译器、宏汇编器、连接/定位器、目标代码到 HEX 的转换器。 μ Vision3 通过以下特性加速用户嵌入式系统的开发过程。

- ⑤ 全功能的源代码编辑器;
- ⑤ 器件库用来配置开发工具设置;
- ⑤ 项目管理器用来创建和维护用户的项目;
- ⑤ 集成的 MAKE 工具可以汇编、编译和连接用户嵌入式应用;
- ⑤ 所有开发工具的设置都是对话框形式的;
- ⑤ 真正的源代码级的对 CPU 和外围器件的调试器;

- ☺ 高级 GDI (AGDI) 接口用来在目标硬件上进行软件调试以及和 Monitor-51 进行通信;
- ☺ 与开发工具手册、器件数据手册和用户指南有直接的链接。

1. μ Vision3 开发环境

运行 μ Vision3 程序, 将出现程序启动界面, 如图 4-84 所示。之后, 程序进入 μ Vision3 用户界面主窗口, 如图 4-85 所示。

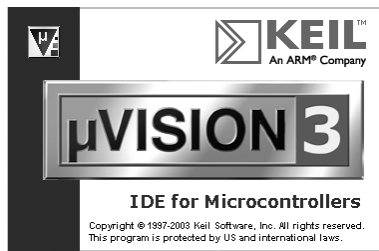


图 4-84 启动界面

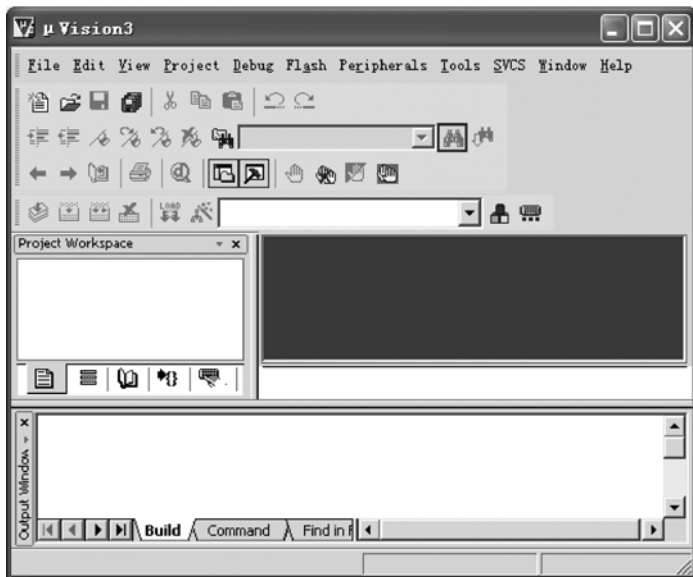


图 4-85 μ Vision3 用户界面主窗口

主窗口提供一个菜单、一个工具条, 以使用户快速选择命令按钮、源代码的显示窗口、对话框和信息显示。 μ Vision3 允许同时打开浏览多个源文件。

2. 建立应用


采用 Keil C51 开发 8051 单片机应用程序一般需要以下步骤:

- (1) 在 μ Vision3 集成开发环境中创建一个新项目文件 (Project), 并为该项目选定合适的单片机 CPU 器件。
- (2) 利用 μ Vision3 的文件编辑器编写 C 语言 (或汇编语言) 源程序文件, 并将文件添加到项目中去。一个项目可以包含多个文件, 除源程序文件外还可以有库文件或文本说明文件。
- (3) 通过 μ Vision3 的各种选项, 配置 C51 编译器、A51 宏汇编器、BL51 连接定位器以及 Debug 调试器的功能。
- (4) 利用 μ Vision3 的构造 (Build) 功能对项目中的源程序文件进行编译连接, 生成绝对目标代码和可选的 HEX 文件。如果出现编译连接错误则返回第 (2) 步, 修改源程序中的错误后重新构造整个项目。
- (5) 将没有错误的绝对目标代码装入 μ Vision3 调试器进行仿真调试, 调试成功后将 HEX 文件写到单片机应用系统的 EPROM 中。

3. 创建项目

μVision3 具有强大的项目管理功能, 一个项目由源程序文件、开发工具选项以及编程说明三部分组成, 通过目标创建 (Build Target) 选项很容易实现对一个 μVision3 项目进行完整的编译连接, 直接产生最终应用目标程序。

(1) 双击 Keil μVision3 图标, 启动应用程序, 进入 μVision3 用户界面主窗口。

μVision3 提供下拉菜单和快捷工具按钮两种操作方法。新建一个源文件时可以通过单击工具按钮图标, 也可以通过选择菜单 File→New 命令, 单击选项后将在项目窗口中打开一个新的文本窗口, 即 Text1 源文件编辑窗口, 如图 4-86 所示。

在该窗口中可以进行源程序文件的编辑, 还可从键盘输入 C 源程序、汇编源程序、混合语言源程序, 源程序输入完毕, 保存文件, 选择菜单 File→Save as 命令, 出现如图 4-87 所示的对话框, 单击“保存”按钮。本例中文件保存为 Text1.a。

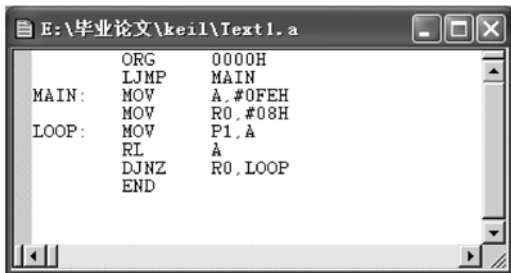


图 4-86 Text1 源文件编辑窗口

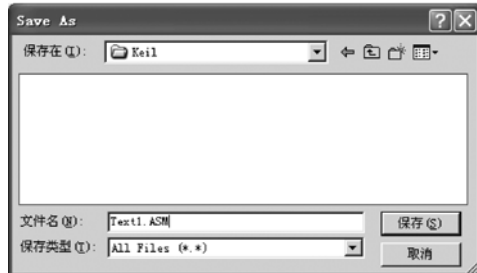


图 4-87 保存源文件窗口

注意:

源程序文件必须加上扩展名 (*.c, *.h, *.a*, *.inc, *.txt)。源程序文件就是一般的文本文件, 不一定使用 Keil 软件编写, 可以使用任何文本编辑器编写。可把源文件, 包括 Microsoft Word 文件中的源文件复制到 Keil C51 文件窗口中, 使 Word 文档变为 TXT 文档。这种方法最好, 可方便对源文件输入中文注释。

(2) 创建一个项目。源程序文件编辑好后, 要进行编译、汇编、连接。Keil C51 软件只能对项目而不能对单一的源程序进行编译、汇编、连接等操作。μVision3 集成环境提供了强大的项目 (Project) 管理功能, 通过项目文件可以方便地进行应用程序的开发。一个项目中可以包含各种文件, 如源程序文件、头文件、说明文件等。因此, 当源文件编辑好后, 要为源程序建立项目文件。

以下是新建一个项目文件的操作。单击 Project→New Project 命令, 弹出一个标准的 Windows 对话框, 此对话框要求输入项目文件名; 输入项目文件名 max (不需要扩展名), 并选择合适的保存路径 (通常为每个项目建立一个单独的文件夹), 单击“保存”按钮, 这样就创建了文件名为 max.uv2 的新项目, 如图 4-88 所示。

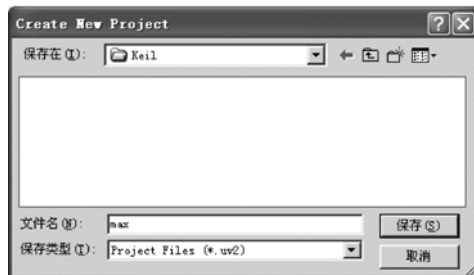


图 4-88 在 μVision3 中新建一个项目

项目文件名保存完毕后,弹出如图 4-89 所示的器件数据库对话框窗口,用于为新建项目选择一个 CPU 器件。Keil C51 支持的 CPU 器件很多,在选择对话框中选 Atmel 公司的 AT89C51 芯片。选定 CPU 器件后,μVision3 按所选器件自动设置默认的工具选项,从而简化了项目的配置过程。选好器件后单击“确定”按钮,此时弹出对话框,询问是否“复制标准 8051 启动代码到工程文件夹,并添加文件到工程?”,如图 4-90 所示。

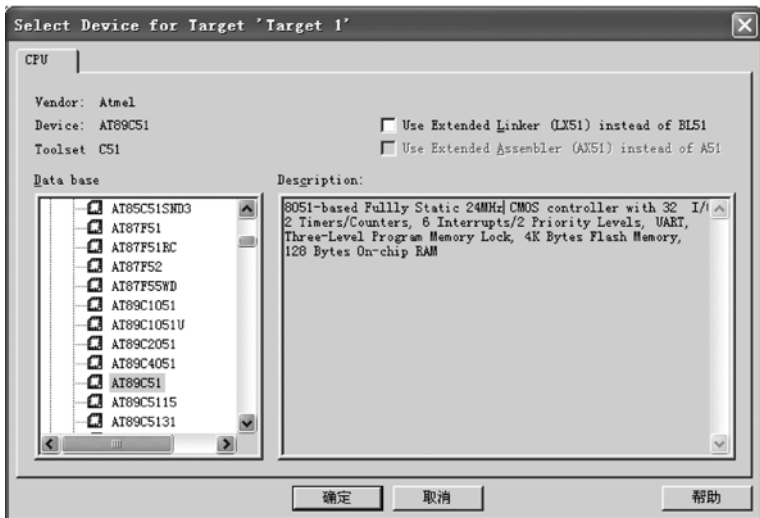


图 4-89 为项目选择 CPU 器件



图 4-90 工程创建提示信息

单击“是”按钮完成项目的新建。创建一个新项目后,项目中会自动包含一个默认的目标(Target 1)和文件组(Source Group 1)。用户可以给项目添加其他项目组(Group)以及文件组的源文件,这对于模块化编程特别有用。项目中的目标名、组名以及文件名都显示在 μVision3 的“项目窗口/File”标签页中。

μVision3 具有十分完善的右键功能,将鼠标指向“项目窗口/File”标签页中的 Source Group 1 文件组并右击,弹出快捷菜单,如图 4-91 所示。

用左键单击右键快捷菜单中的 Add Files to Group ‘Source Group 1’选项,弹出如图 4-92 所示的添加源文件选择窗口,选择待添加的源文件。

注意:

该对话框下面的“文件类型”默认为.c(C语言源程序),而待添加的文件是以.asm(汇编语言源程序)为扩展名的,所以要将对话框下面的文件类型进行修改。单击对话框中的“文件类型”后的下拉式列表,找到并选中“Asm Source file(*.a, *.sor)”选项,这样,在列表中就可以找到 Text1.asm 文件。双击 Text1.asm 文件,就可以将汇编语言文件加到新创建的项目中去。

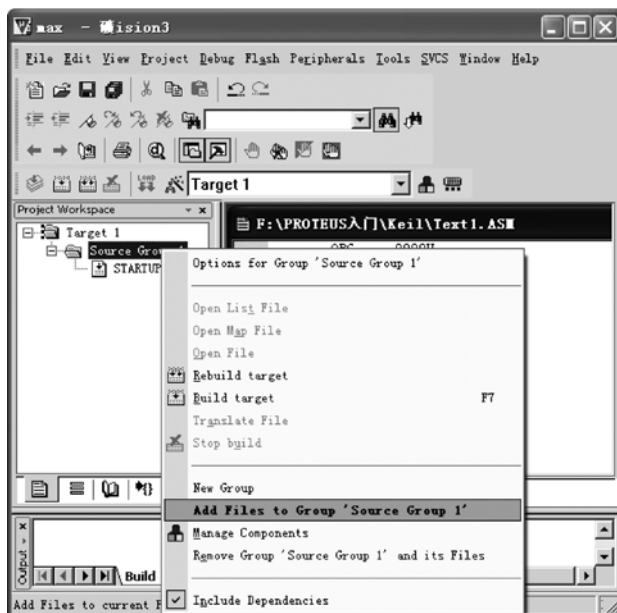


图 4-91 项目窗口的右键菜单

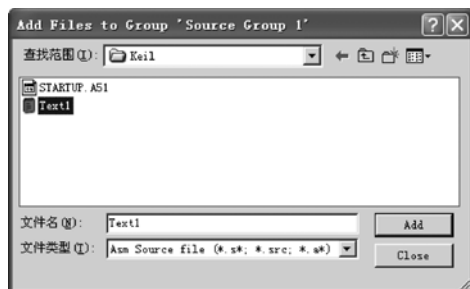


图 4-92 添加源文件选择窗口

4. 项目的设置

项目建立好后,还要根据需要设置项目目标硬件 C51 编译器、A51 宏汇编器、BL51 连接定位器以及 Debug 调试器的各项功能。单击 Project→Options for Target ‘Target 1’选项,弹出如图 4-93 所示窗口。



图 4-93 Options 选项中的 Target 标签页

注意:

这是一个十分重要的窗口,包括“Target”、“Output”、“Listing”、“C51”、“A51”、“BL51 Locate”、“BL51 Misc”、“Debug”选项标签页,其中许多选项可以直接用其默认值,必要时可进行适当调整。

5. 项目的编译、链接

设置好项目后,即可对当前项目进行整体创建。将鼠标指向项目窗口中的文件 Text1.asm 并右击,从弹出的快捷菜单中选择“Build target”选项,如图 4-94 所示。

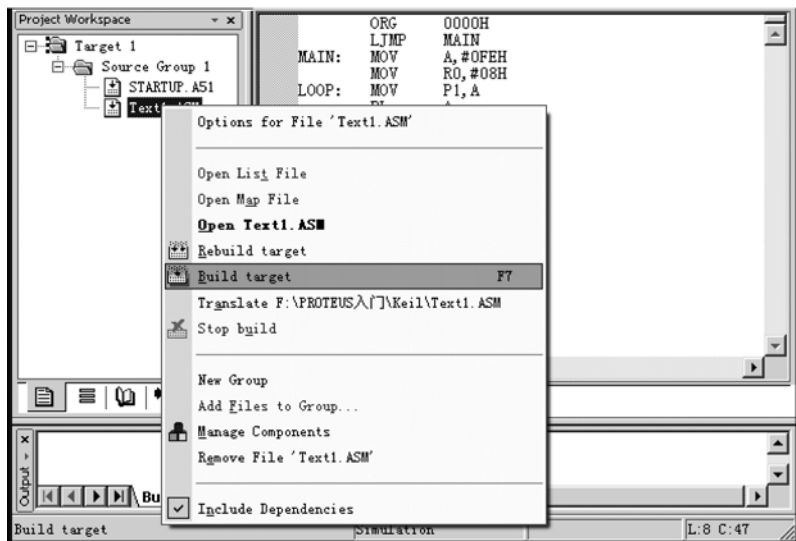


图 4-94 利用右键快捷菜单对当前项目进行编译、链接

μ Vision3 将按“Options for Target”窗口内的各种选项设置,自动完成对当前项目中所有源程序模块的编译、链接。

同时 μ Vision3 的输出窗口将显示编译、链接过程中的提示信息,如图 4-95 所示。

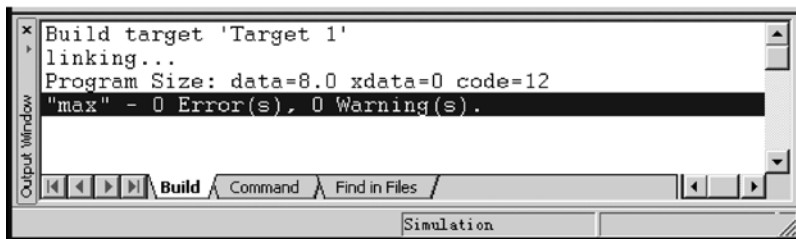


图 4-95 编译提示信息

注意:

如果源程序中有语法错误,将鼠标指向窗口内的提示信息双击,光标将自动跳到编辑窗口源程序出错位置,以便于修改;如果没有编译错误,则生成绝对目标代码文件。

6. 程序调试

在对项目成功地进行汇编、链接以后,将 μ Vision3 转入仿真调试状态,单击选择菜单 Debug→Start/Stop Debug Session 命令,即可进入调试状态,如图 4-96 所示。在此状态下的项目窗口自动转换到“Regs”标签页,显示调试过程中单片机内部工作寄存器 R0~R7、累加器 A、堆栈指针 SP、数据指针 DPTR、程序计数器 PC 以及程序状态字 PSW 等的值。

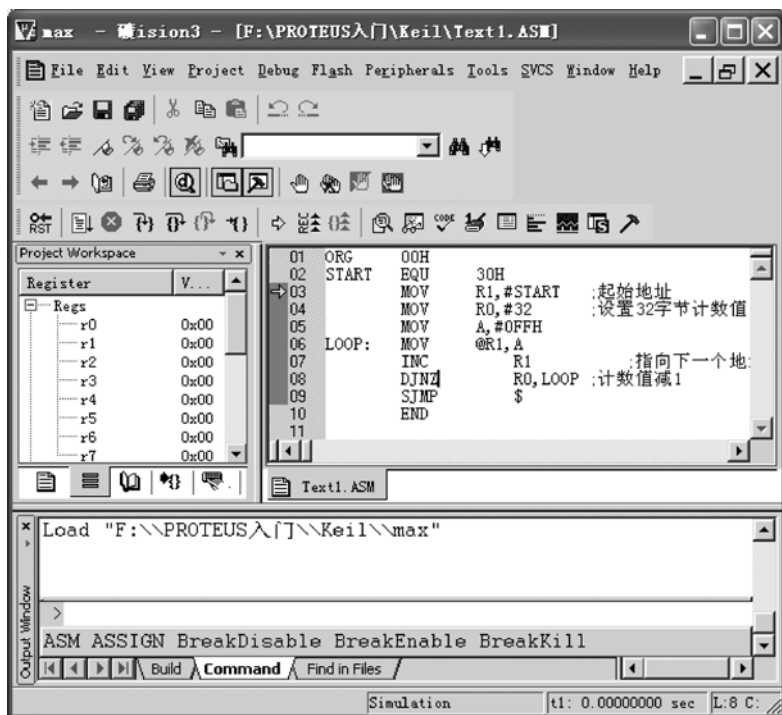


图 4-96 μVision3 仿真调试状态窗口

在仿真调试状态下，单击选择菜单 Debug→Run 命令，启动用户程序全速运行，如图 4-97 所示。

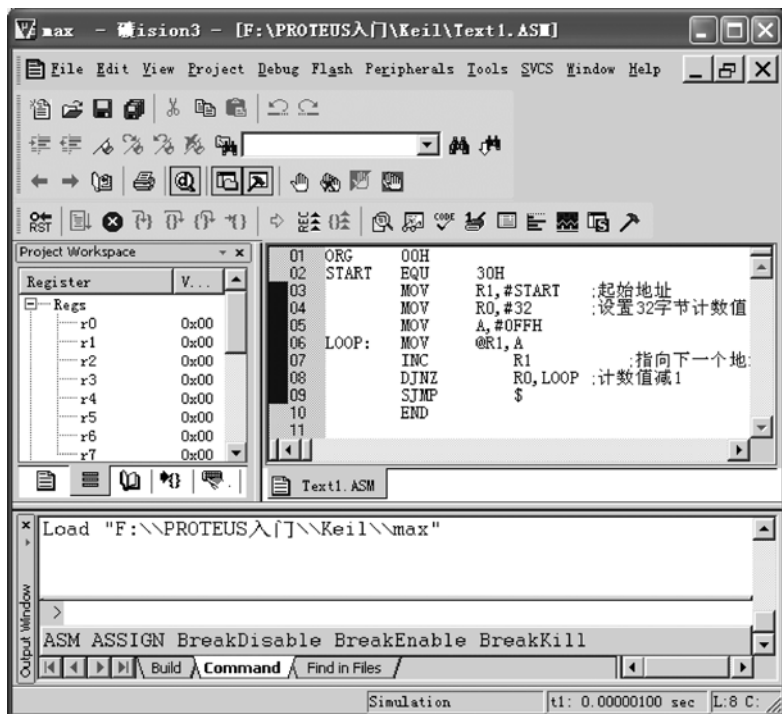


图 4-97 用户程序运行输出窗口

图 4-98 所示为模拟调试窗口的工具栏快捷按钮。

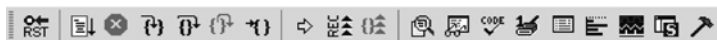


图 4-98 μVision3 调试工具按钮

“Debug”下拉式菜单上的大部分选项可以在此找到对应的快捷按钮。工具栏快捷按钮的功能从左到右依次为：复位、运行、暂停、单步、过程单步、执行完当前子程序、运行到当前行、下一状态、打开跟踪、查看跟踪、反汇编窗口、观察窗口、代码作用范围分析、1# 串行窗口、内存窗口、性能分析、逻辑分析窗口、符号窗口及工具按钮。

7. 在线汇编

在进入 Keil 的调试环境以后，如果发现程序有错误，可以直接对源程序进行修改。但是要使用修改后的代码起作用，必须先退出调试环境，重新进行编译、链接后再进入调试。这样的过程未免有些麻烦。为此，Keil 软件提供了在线汇编的功能：将光标定位于需要修改的程序语句上，点选 Debug→Inline Assembly 命令，如图 4-99 所示。

此时将出现如图 4-100 所示的 Debug 菜单在线汇编的功能窗口。

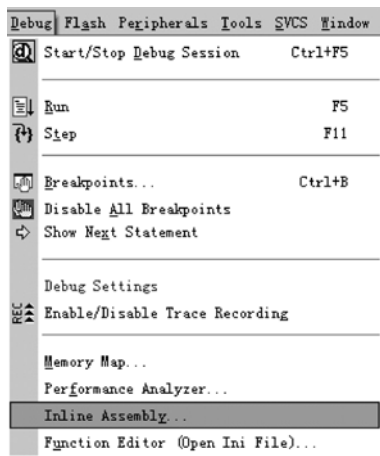


图 4-99 Debug→Inline Assembly 命令

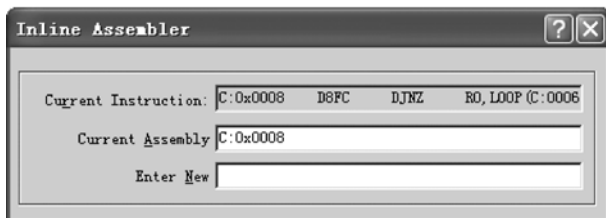


图 4-100 Debug 菜单在线汇编的功能窗口

在 Enter New 文本框内直接输入需要更改的程序语句，输入完成后按回车键，程序将自动指向源程序的下一条语句，继续修改；如果不需要继续修改，可以单击窗口右上角的关闭按钮，关闭窗口。

8. 断点管理

断点功能对于用户程序的仿真调试是十分重要的，利用断点调试，便于观察了解程序的运行状态，查找或排除错误。Keil 软件在 Debug 调试命令菜单中设置断点的功能。在程序中设置、移除断点的方法是：在汇编窗口光标定位于需要设置断点的程序行，单击 Debug→Insert/Remove Breakpoint 命令，可在编辑窗口当前光标所在行上设置/移除一个断点（也可用鼠标在该行双击实现同样功能）；单击 Debug→Enable/Disable Breakpoint 选项，可激活/禁止当前光标所指向的一个断点；单击 Debug→Disable All Breakpoint 选项，将禁止所有的已经设置的断点；单击 Debug→Kill All Breakpoint 选项，将清除所有已经设置的断点；单击 Debug→

Show Next Statement 选项, 将在汇编窗口显示下一条将要被执行的用户程序指令。

除了在程序行上设置断点这一基本方法外, Keil 软件还提供了通过断点设置窗口 (Breakpoints) 来设置断点的方法。单击 Debug→Breakpoints 选项, 将弹出如图 4-101 所示的对话框。

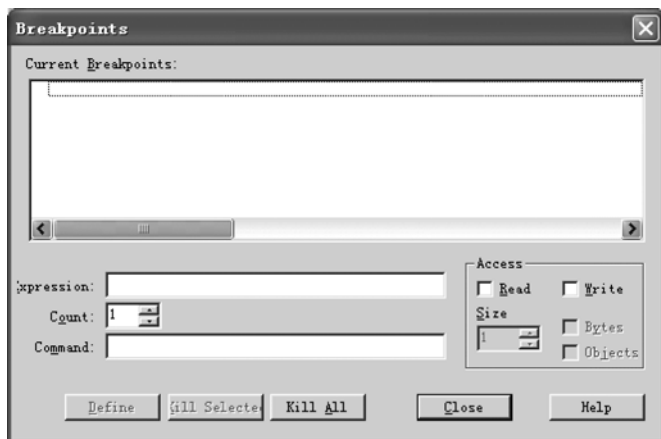


图 4-101 断点设置对话框

该对话框用于对断点进行详细设置。窗口中 Current Breakpoints 栏显示当前已经设置的断点列表; 窗口中 Expression 栏用于输入断点表达式, 该表达式用于确定程序停止运行的条件; Count 栏用于输入断点通过的次数; Command 用于输入当程序执行到断点时需要执行的命令。

9. Keil 的模拟仿真调试窗口

Keil 软件在对程序进行调试时提供了多个模拟仿真窗口, 主要包括主调试窗口、输出调试窗口 (Output Window)、观测窗口 (Watch & Call Stack Window)、存储器窗口 (Memory Window)、反汇编窗口 (Disassembly Window)、串行窗口 (Serial Window) 等。进入调试模式后, 通过单击 View 菜单中的相应选项 (或单击工具条中的相应按钮), 可以很方便地实现窗口的切换。

调试状态下的 View 菜单如图 4-102 所示。第一栏用于快捷工具条按钮的显示/隐藏切换。Status Bar 选项为状态栏; File Toolbar 选项为调试工具条按钮。第二栏、第三栏用于 μ Vision3 中各种窗口的显示/隐藏切换。

【存储器窗口】 View 菜单的 Memory Window 选项用于系统存储器空间的显示/隐藏切换, 如图 4-103 所示。

存储器窗口用于显示程序调试过程中单片机的存储器系统中各类存储器中的值, 在窗口 Address 文本框中输入存储器地址 (字母: 数字), 将立即显示对应存储空间的内容。

注意:

输入地址时要指定存储器的类型 C、D、I、X 等, 其含义分别是: C 为代码 (ROM) 存储空间; D 为直接寻址的片内存储空间; I 为间接寻址的片内存储空间; X 为扩展的外部 RAM 空间。数字的含义为要查看的地址值。例如输入 D: 0, 可查看地址 0 开始的片内 RAM 单元的内容; 输入 C: 0, 可查看地址 0 开始的 ROM 单元中的内容, 也就是查看程序的二进制代码。



图 4-102 调试状态下的 View 菜单

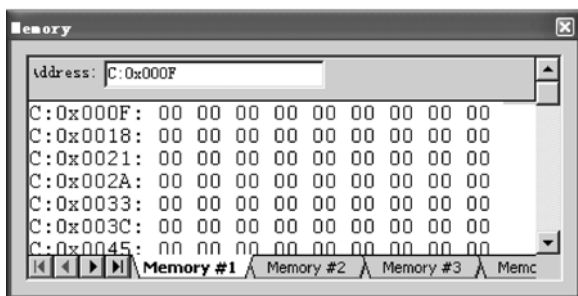


图 4-103 存储器窗口

存储器窗口的显示值可以是十进制、十六进制、字符型等多种形式，改变显示形式的方法是：在存储器窗口单击鼠标右键，弹出如图 4-104 所示的快捷菜单，用于改变存储器内容的显示方式。

【观测窗口】观测窗口也是调试程序中的一个重要窗口，在项目窗口中仅可以观察到工作寄存器和有限的寄存器内容，如寄存器 A、B、DPTR 等，若要观察其他寄存器的值或在高级语言程序调试时直接观察变量，则需要借助于观测窗口。单击工具栏上观测窗口的快捷按钮可打开观测窗口。观测窗口有四个标签，分别是局部变量（Locals）、观测 1（Watch#1）、观测 2（Watch#2）以及调用堆栈（Call Stack）标签页。图 4-105 所示为观测窗口的局部变量 Locals 页，显示用户调用程序的过程中当前局部变量的使用情况。

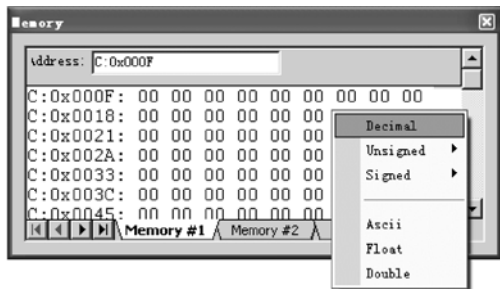


图 4-104 存储器窗口右键菜单

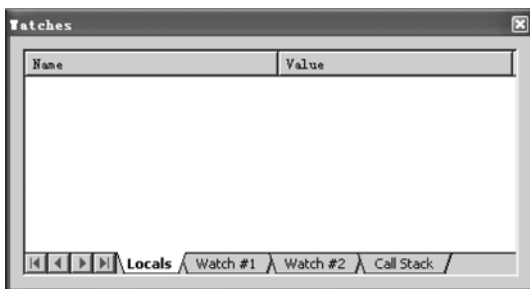


图 4-105 观测窗口的局部变量 Locals 页

图 4-106 所示为观测窗口的 Watch#1 页，显示用户程序中已经设置了的观测点在调试中的当前值；在 Locals 栏和 Watch#1 栏中单击鼠标右键，可改变局部变量或观测点的值按十六进制（Hex）或十进制（Decimal）方式显示。

图 4-107 所示为观测窗口的 Call Stack 页，显示程序执行过程中对子程序的调用情况。

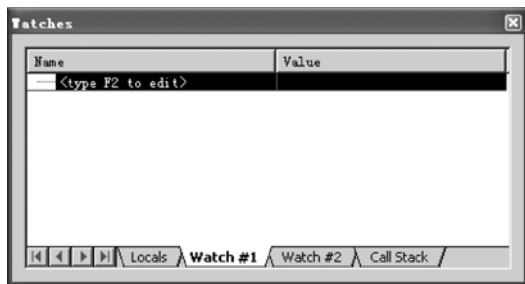



图 4-106 观测窗口的 Watch#1 页



图 4-107 观测窗口的 Call Stack 页

另外，单击 View→Periodic Window Update（周期更新窗口）命令，可在用户程序全速运行时动态地观察程序中相关变量值的变化。

【项目窗口寄存器页】项目窗口在仿真调试状态下自动转换到 Regs（寄存器）标签页。在调试中，当程序执行到对某个寄存器操作时，该寄存器会以反色（蓝底白字）显示。用鼠标左键单击窗口某个寄存器然后按 F2 键，即可修改寄存器的内容。

【反汇编窗口】选中 View→Disassembly Window 命令，或单击调试工具条上的反汇编快捷图标按钮，可打开如图 4-108 所示的反汇编窗口，用于显示已装入到 μVision3 的用户程序汇编语言指令、反汇编代码及其地址。

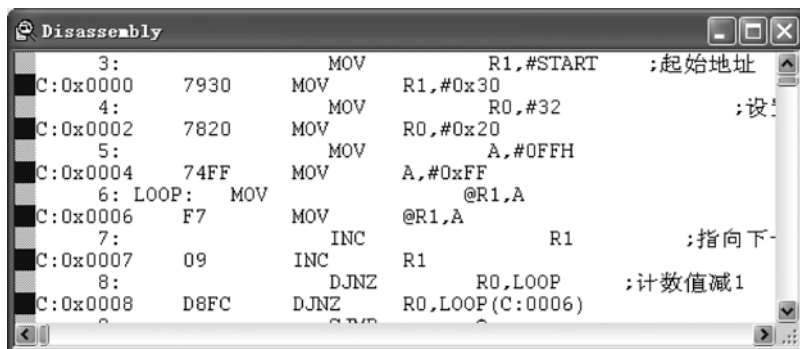


图 4-108 反汇编窗口

当采用单步或断点方式运行程序时，反汇编窗口的显示内容会随指令的执行而滚动。在反汇编窗口中可以使用右键功能，方法是将鼠标指向反汇编窗口并单击右键，可弹出如图 4-109 所示的窗口。

该窗口第一栏中的选项用于选择窗口内反汇编内容的显示方式，其中 Mixed Mode 选项采用高级语言与汇编语言混合方式显示；Assembly Mode 选项采用汇编语言方式显示；Inline Assembly 选项用于程序调试中“在线汇编”，利用窗口跟踪已执行的代码。

右键窗口第二栏的 Address Range 选项用于显示用户程序的地址范围；Load Hex or Object file 选项用于重新装入 Hex 或 Object 文件到 μVision3 中调试。

右键窗口第三栏的 View Trace Records 选项用于在反汇编窗口显示指令执行的历史记录，该选项只有在该栏中另一选项 Enable/Disable Trace Recording 被选中，并已经执行过用户程序指令的情况下才起作用；选项 Show next statement 用于显示下一条指令；Run till

Cursor line 选项用于将程序执行到当前光标所在的那一行；Insert/Remove Breakpoint 选项用于插入/删除程序执行时的断点；Enable/Disable Breakpoint 选项可以激活/禁止选定一个断点；Clear complete Code Coverage Info 选项用于清零代码覆盖信息。

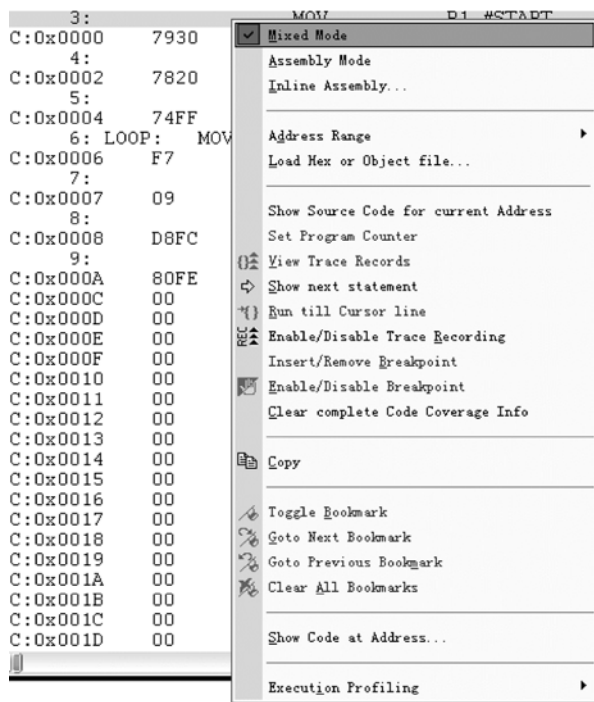


图 4-109 反汇编窗口中右键菜单

右键窗口第四栏的 Copy 选项用于复制反汇编窗口中的内容。

右键窗口的 Show Code at Address 选项用于显示指定地址处的用户程序代码。

【串行窗口】 View→Serial Window #1/ Serial Window #2/ Serial Window #3 选项用于串行窗口 1、串行窗口 2 和串行窗口 3 的显示/隐藏切换，选中该项则弹出串行窗口。串行窗口在进行用户程序调试时十分有用，如果用户程序中调用了 C51 的库函数 scanf() 和 printf()，则必须利用串行窗口来完成 scanf() 函数的输入操作，printf() 函数的输出结果也将显示在串行窗口中。利用串行窗口可以在用户程序仿真调试过程中实现人机交互对话，可以直接在串行窗口中输入字符。该字符不会被显示出来，但却能传递到仿真 CPU 中。如果仿真 CPU 通过串口发送字符，那么，这些字符会在串行窗口显示出来。串行窗口可以在没有硬件的情况下用键盘模拟串口通信。在串行窗口单击鼠标右键将弹出如图 4-110 所示的显示方式选择菜单，可按需要将窗口内容以 Hex 或 ASCII 格式显示，也可以随时清除显示内容。串行窗口中可保持近 8KB 串行输入/输出数据，并可以进行翻滚显示。

Keil 的串行窗口除了可以模拟串行口的输入和输出外，还可以与 PC 上实际的串口相连，接收串口输入的内容，并将信息输出到串口。

【通过 Peripherals 菜单观察仿真结果】 μVision3 通过内部集成器件库实现对各种单片机外围接口功能的模拟仿真，在调试状态下可以通过 Peripherals 下拉式菜单来直观地观察单片机的定时器、中断、并行端口、串行端口等常用外围接口的仿真结果。Peripherals 菜单如图 4-111 所示。

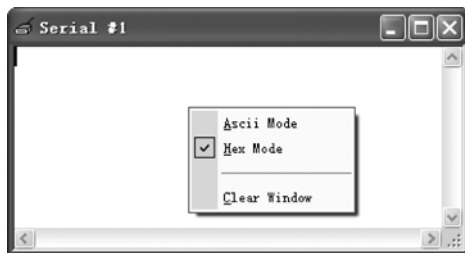


图 4-110 串行窗口显示方式选择菜单

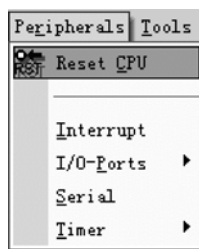


图 4-111 Peripherals 菜单

该下拉式菜单的内容与建立项目时所选的 CPU 器件有关，如果选择的是 89C51 这一类“标准”的 51 机，则有 Interrupt（中断）、I/O-Ports（并行 I/O 口）、Serial（串行口）、Timer（定时/计数器）这四个外围接口菜单选项，打开这些对话框，系统列出了这些外围设备当前的使用情况，以及单片机对应的特殊功能寄存器各标志位的当前状态等。

单击 Peripherals 菜单第一栏 Reset CPU 选项可以对模拟仿真的 8051 单片机进行复位。

Peripherals 菜单第二栏中 I/O-Ports 选项用于仿真 8051 单片机的 I/O 接口 Port 0~Port 3，选中 Port 1 后将弹出如图 4-112 所示的窗口，其中 P1 栏显示 8051 单片机 P1 口锁存器状态，Pins 栏显示 P1 口各引脚状态。

Peripherals 菜单最后一栏 Timer 选项用于仿真 8051 单片机内部定时/计数器。选中其中 Timer 0 后弹出如图 4-113 所示的窗口。

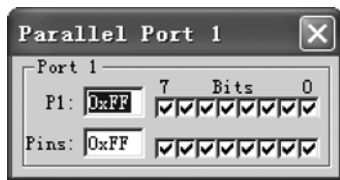


图 4-112 Port 1 窗口



图 4-113 Timer 0 窗口

窗口中 Mode 栏用于选择工作方式，可选择定时/计数器工作方式，单击其中的箭头很容易实现选择，图 4-113 所示为 13 位定时器工作方式。选定工作方式后相应的特殊寄存器 TCON 和 TMOD 控制字也显示在窗口中，可以直接写入命令字；窗口中的 TH0 和 TL0 项用于显示定时/计数器 0 的定时/计数初值；T0 Pin 和 TF0 复选框用于显示 T0 引脚和定时/计数器 0 的溢出状态。窗口中的 Control 栏用于显示和控制定时/计数器 0 的工作状态（Run 或 Stop），TR0、GATE、INT0#复选框是启动控制位，通过对这些状态位的置位或复位操作（选中或不选中），很容易实现对 8051 单片机内部定时/计数器的仿真。单击 TR0，启动定时/计数器 0 开始工作，这时 tatus 文本框内的 Stop 变成 Run。如果全速运行程序，可观察到 TH0、TL0 后的值也在快速变化。当然，由于上述源程序未对对话框写入任何信息，所以该程序运行时不会对定时/计数器 0 的工作进行处理。

Peripherals 菜单第二栏中 Serial 选项用于仿真 8051 单片机的串行口。单击该选项弹出如图 4-114 所示的窗口。

窗口中 Mode 栏用于选择串行口的工作方式，选定工作方式后相应的特殊寄存器 SCON 和 SBUF 的控制字也显示在窗口中。通过对特殊控制位 SM2、REN、TB8、RB8、TI、RI 复选框的置位或复位操作（选中或不选中），很容易实现对 8051 单片机内部串行口的仿真。Baudrate 栏用于显示串行口的工作波特率，SMOD 位置位时将使波特率加倍。IRQ 栏用于显示串行口发送和接收中断标志。

Peripherals 菜单第二栏中的 Interrupt 选项用于仿真 8051 单片机的中断系统状态。单击该选项弹出如图 4-115 所示的窗口。



图 4-114 串行口窗口

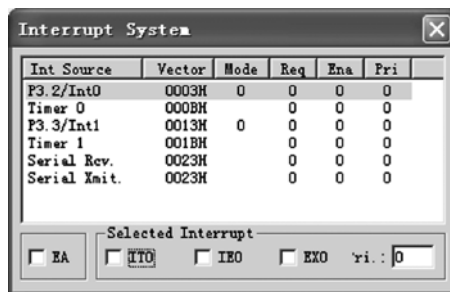


图 4-115 系统中断窗口

选中不同的中断源，窗口中的 Selected Interrupt 栏将出现与之相对应的中断允许和中断标志位的复选框，通过对这些标志位的置位或复位操作（选中或不选中），很容易实现对 8051 单片机中断系统的仿真。除了 8051 几个基本的中断源以外，还可以对其他中断源如看门狗定时器（Watchdog Timer）等进行模拟仿真。

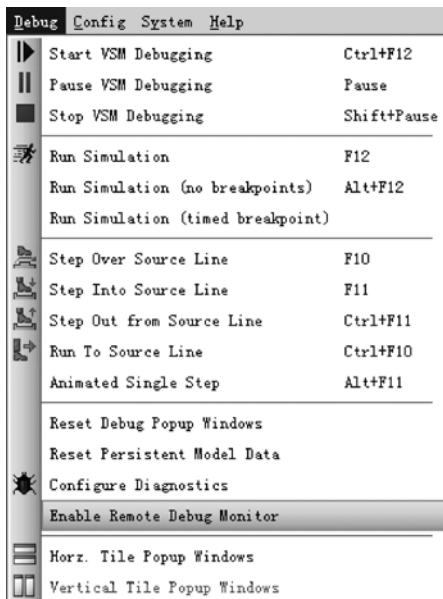


图 4-116 选择 Debug→Enable Remote Debug Monitor 命令

4.4.2 进行 PROTEUS 与 Keil 的整合

在 Keil 中调用 PROTEUS 进行 MCU 外围器件的仿真的步骤如下：

- (1) 安装 Keil 与 PROTEUS 软件。
- (2) 安装 Keil 与 PROTEUS 软件的链接文件 vdmagdi.exe。
- (3) 打开 PROTEUS，画出相应电路，选择 Debug→Enable Remote Debug Monitor 命令，如图 4-116 所示。
- (4) 在 Keil 中编写 MCU 的程序。
- (5) 在 Keil 中选择 Project→Options for Target ‘工程名’ 命令，如图 4-117 所示。
- (6) 在弹出的对话框中，點選 Debug 选项中右栏上部的下拉菜单，选中 Proteus VSM Simulator 选项，如图 4-118 所示。

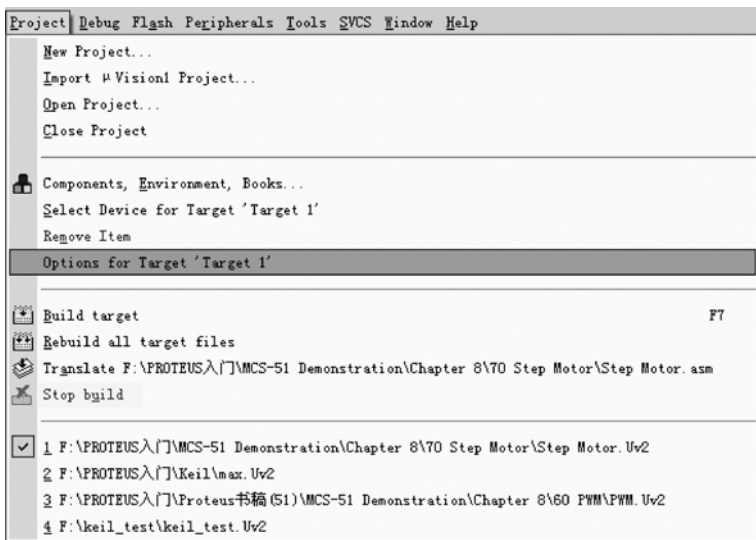



图 4-117 选择 Project→Options for Target ‘工程名’ 命令



图 4-118 在 Debug 中选择 Proteus VSM Simulator

单击“确定”按钮完成设置。

单击 Keil 中的启动调试按钮, 此时 Keil 与 PROTEUS 实现联调。

4.4.3 进行 PROTEUS 与 Keil 的联调

本节以存储块清零为例说明 PROTEUS 与 Keil 联调的过程。

说明:

存储块清零指定某块存储空间的起始地址和长度, 要求能将存储器内容清零。通过该实验, 可以了解单片机读写存储器的方法, 同时也可以了解单片机编程、调试方法。

1. 程序流程

如图 4-119 所示, 源程序为:

```
ORG      00H
```

```

START    EQU      30H
        MOV      R1,#START    ; 起始地址
        MOV      R0,#32      ; 设置 32 字节计数值
        MOV      A,#0FFH
LOOP:    MOV      @R1,A
        INC      R1          ; 指向下一个地址
        DJNZ     R0,LOOP      ; 计数值减 1
        SJMP     $
        END

```

2. 在 Keil 中调试程序

打开 Keil μ Vision3, 在菜单栏中选择 Project→New Project, 弹出 Create New Project 对话框, 选择目标路径, 在“文件名”栏中输入项目名, 如图 4-120 所示。

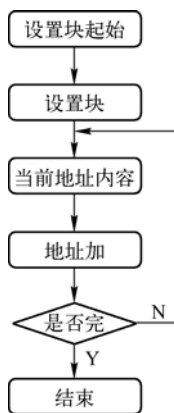


图 4-119 存储块清零源程序

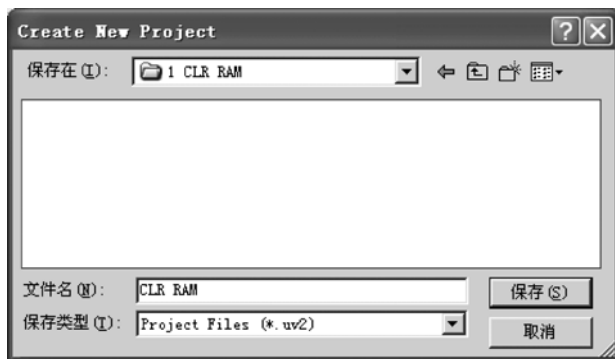


图 4-120 新建项目

单击“保存”按钮, 这时会弹出 Select Device for Target 对话框。在此对话框的 Data base 栏中单击 Atmel 前面的+号, 或者直接双击 Atmel, 在其子类中选择 AT89C51 芯片, 确定 CPU 类型, 如图 4-121 所示。

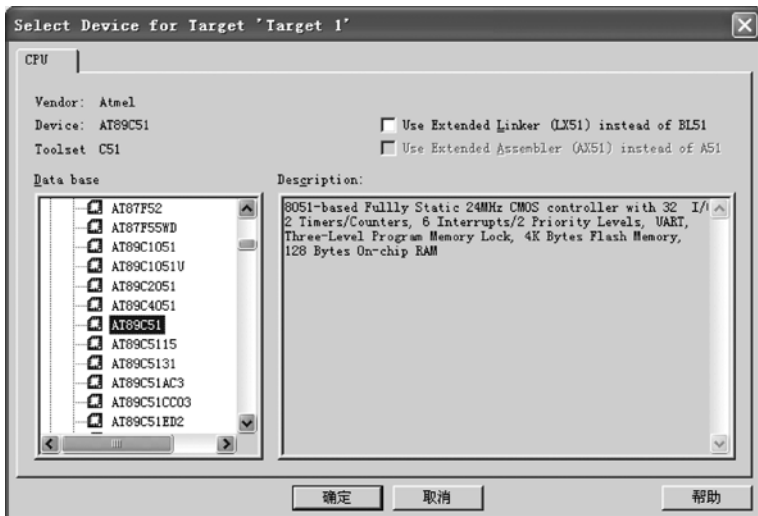


图 4-121 选择 CPU

在 Keil μ Vision3 的菜单栏中选择 File→New，新建文档，然后在菜单栏中选择 File→Save，保存此文档，这时会弹出 Save As 对话框，在“文件名”一栏中为此文本命名，如图 4-122 所示。

注意：

要填写扩展名.asm。

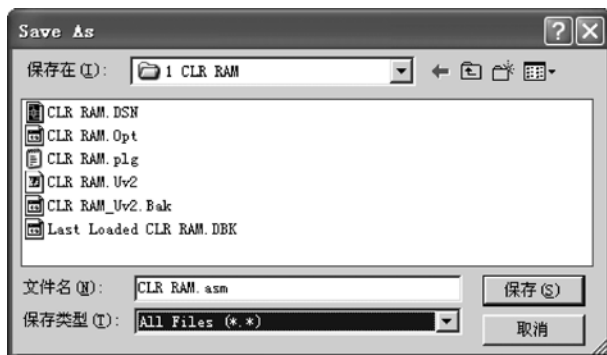


图 4-122 保存文本

单击“保存”按钮，这样在编写汇编代码时，Keil 会自动识别汇编语言的关键字，并以不同的颜色显示，以减少在输入代码时出现的语法错误。

程序编写完后，再次保存。在 Keil 中 Project Workspace 子窗口中，单击 Target 1 前的+号，展开此目录。在 Source Group 1 文件夹上单击右键，在右键菜单中选择 Add File to Group ‘Group Source 1’，弹出 Add File to Group 对话框，在此对话框的“文件类型”栏中选择 Asm Source File，并找到刚才编写的.asm 文件，双击此文件，将其添加到 Source Group 中，此时的 Project Workspace 子窗口如图 4-123 所示。

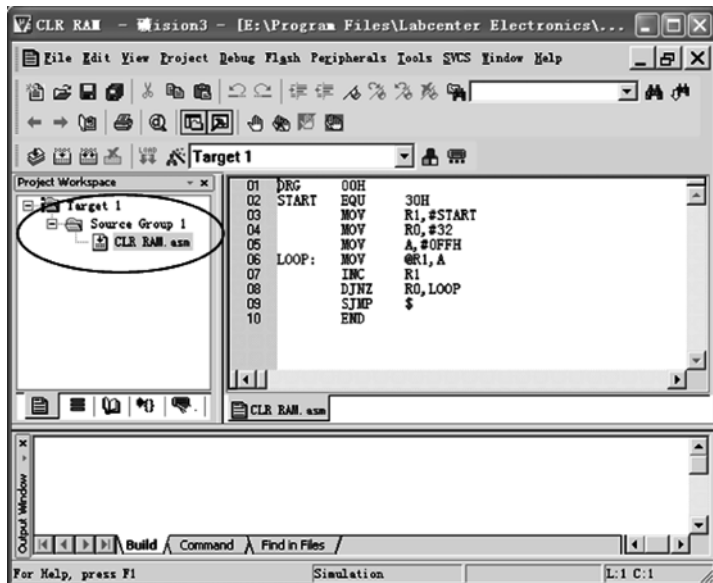


图 4-123 添加源程序

在 Project Workspace 窗口中的 Target 1 文件夹上单击鼠标右键，在弹出的右键菜单中选择 Options for Target 选项，这时会弹出 Options for Target 对话框，在此对话框中选择 Output 选项卡，选中 Create HEX File 选项，如图 4-124 所示。

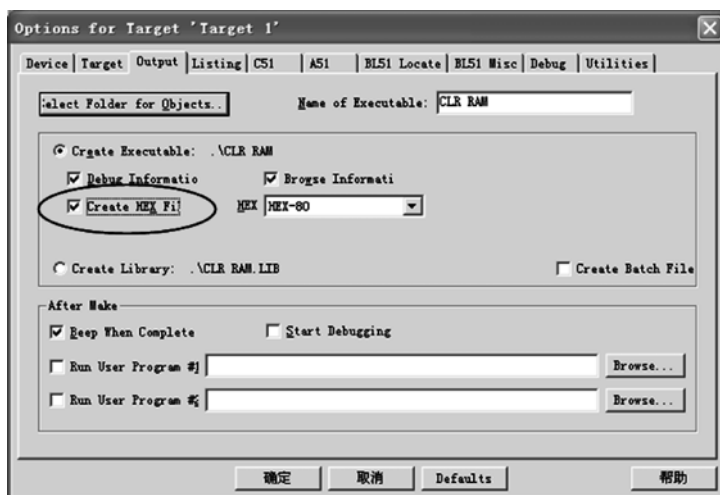


图 4-124 Options for Target 对话框

在 Keil 的菜单栏中选择 Project→Build Target，编译汇编源文件，如果编译成功，则在 Keil 的 Output Window 子窗口中会显示如图 4-125 所示的信息；如果编译不成功，双击 Output Window 窗口中的错误信息，则会在编辑窗口中指示错误的语句。

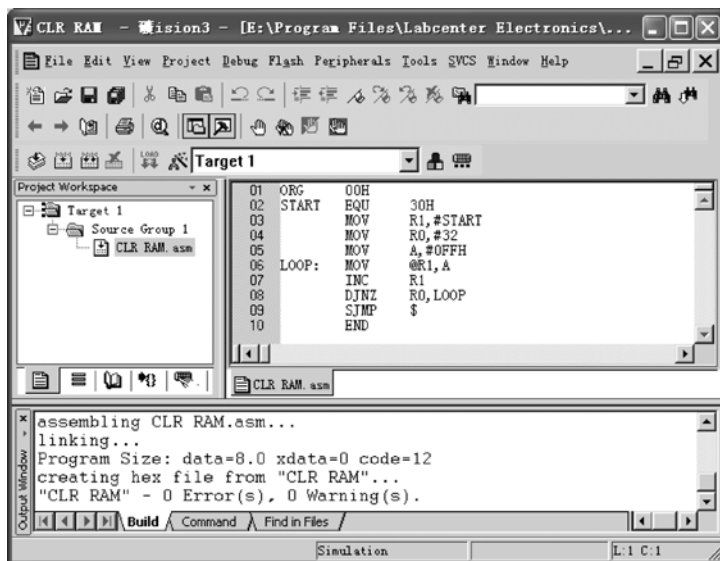


图 4-125 编译源文件

说明：

为了查看程序运行的结果，在这里我们把源程序中第 5 行的语句改写为：

```
MOV    A, #0FFH
```

即把存储空间清零的操作改为置 1 操作，原理相同。

在 Keil 的菜单栏中, 选择 Debug→Start/Stop Debug Session 选项, 进入程序调试环境, 如图 4-126 所示。按 F11 键, 单步运行程序。在 Project Workspace 窗口中, 可以查看累加器、通用寄存器以及特殊功能寄存器的变化; 在 Memory 窗口中, 可以看到每执行一条语句后存储空间的变化。在 Address 栏中输入 D: 30H, 查看 AT89C51 的片内直接寻址空间, 并单步运行程序。可以看到, 随着程序的顺序执行, 从 30H~4FH 这 32 个存储单元依次被置 1。程序调试完毕后, 再次在菜单栏中选择 Debug→Start/Stop Debug Session 选项, 退出调试环境。

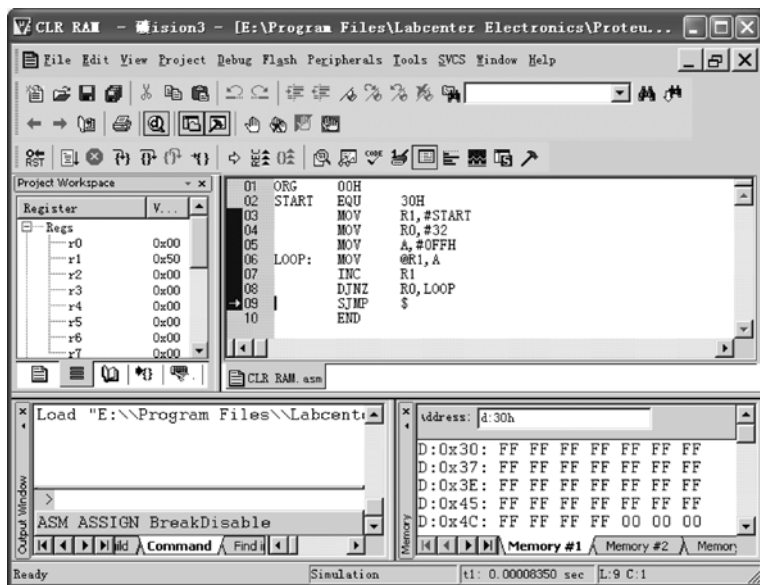


图 4-126 Keil 的程序调试环境

3. 在 PROTEUS 中调试程序

打开 PROTEUS ISIS 编辑环境, 添加器件 AT89C51。

注意:

在 PROTEUS 中添加的 CPU 一定要与 Keil 中选择的 CPU 相同, 否则无法执行 Keil 生成的 .hex 文件。

按照图 4-127 所示连接晶振和复位电路, 晶振频率为 12MHz。元件清单如表 4-2 所示。

选中 AT89C51 并双击左键, 打开 Edit Component 对话框, 在此窗口中的 Program File 栏中选择先前用 Keil 生成的 .hex 文件, 如图 4-128 所示。

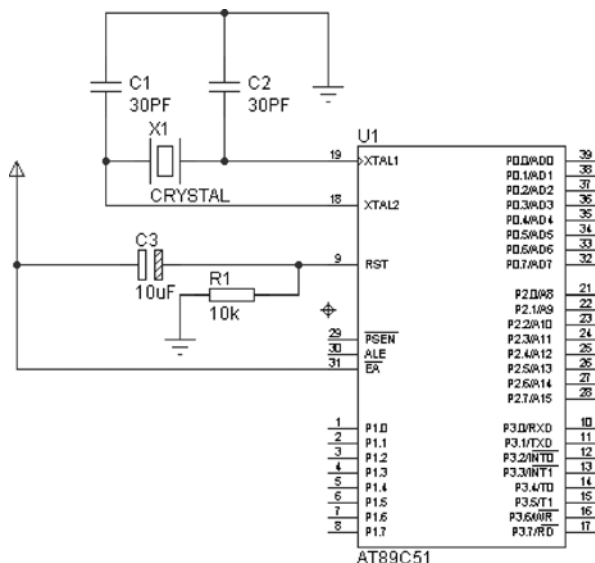


图 4-127 单片机晶振和复位电路

表 4-2 元件清单

元 件 名 称	所 属 类	所 属 子 类
AT89C51	Microprocessor ICs	8051 Family
CAP	Capacitors	Generic
CAP-ELEC	Capacitors	Generic
CRYSTAL	Miscellaneous	—
RES	Resistors	Generic

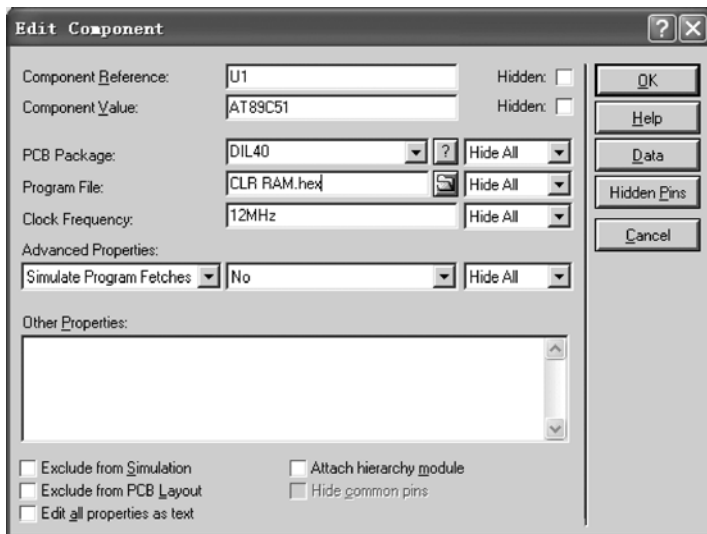



图 4-128 添加.hex 文件

在 PROTEUS ISIS 的菜单栏中选择 File→Save Project，保存设计。在保存设计文件时，最好将与一个设计相关的文件（如 Keil 项目文件、源程序、PROTEUS 设计文件）都存放在一个目录下，以便查找。

单击 PROTEUS ISIS 界面左下角的  按钮，进入程序调试状态，并在 Debug 菜单中打开 8051 CPU Registers、8051 CPU Internal (IDATA) Memory 及 8051 CPU SFR Memory 三个观测窗口，按 F11 键，单步运行程序。在程序运行过程中，可以在这三个窗口中看到各寄存器及存储单元的动态变化。程序运行结束后，8051 CPU Registers 和 8051 CPU Internal (IDATA) Memory 的状态如图 4-129 所示。

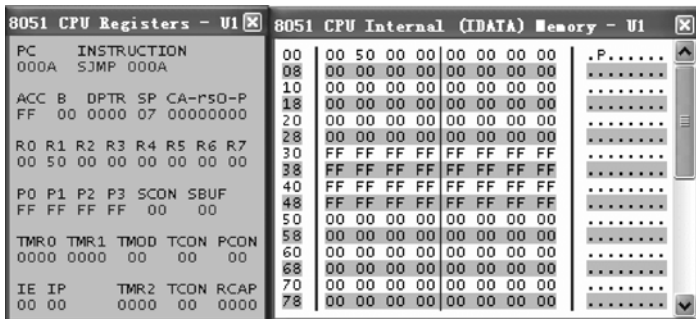



图 4-129 程序运行结果

程序调试成功后，将汇编源程序的第 5 行语句改为：

```
MOV      A, #00H
```

单击 PROTEUS ISIS 界面左下角的  按钮，进入程序调试状态，并在 Debug 菜单中打开 8051 CPU Registers、8051 CPU Internal (IDATA) Memory 及 8051 CPU SFR Memory 三个观测窗口，按 F11 键，单步运行程序。编译后重新运行，在程序运行过程中，即可实现存储块清零的功能，可以在这三个窗口中看到各寄存器及存储单元的动态变化。程序运行结束后，8051 CPU Registers 和 8051 CPU Internal (IDATA) Memory 的状态如图 4-130 所示。

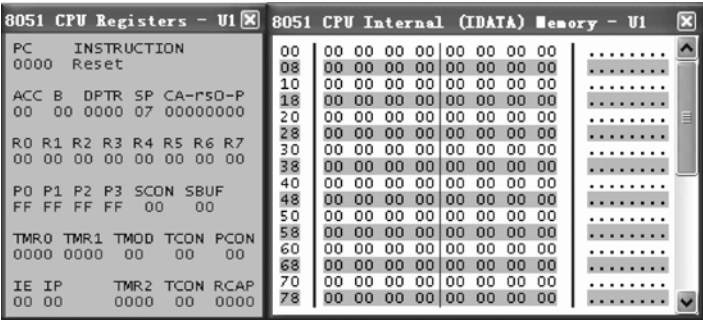


图 4-130 改变语句后的程序运行结果

第 5 章 基于 PROTEUS ISIS 的数字电路分析

PROTEUS ISIS 数字电路分析支持 JDEC 文件的物理器件仿真；有全系列的 TTL 和 CMOS 数字电路仿真模型；可对数字电路进行一致性分析。



5.1 异步十进制计数器电路及其分析

5.1.1 异步十进制计数器电路

根据 8421 码十进制减法计数规则可列出电路的状态转换表，如表 5-1 所示。

表 5-1 十进制减法计数器的状态转换表

计 数 顺 序	电 路 状 态				等效十进制数	输出 B
	Q ₁	Q ₂	Q ₃	Q ₄		
0	0	0	0	0	0	1
1	1	0	0	1	9	0
2	1	0	0	0	8	0
3	0	1	1	1	7	0
4	0	1	1	0	6	0
5	0	1	0	1	5	0
6	0	1	0	0	4	0
7	0	0	1	1	3	0
8	0	0	1	0	2	0
9	0	0	0	1	1	0
10	0	0	0	0	0	1

由表 5-1 可画出如图 5-1 所示的状态转换图。

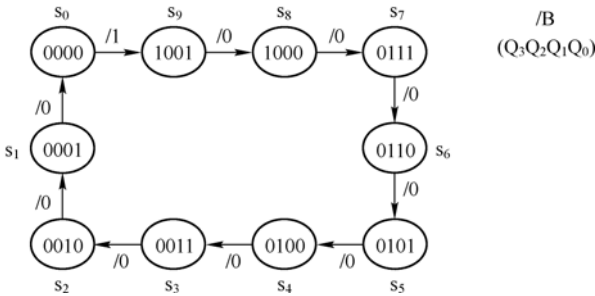


图 5-1 电路的状态转换图

十进制计数器必须有 10 个有效状态, 若依次为 $s_0, s_9, s_8, \dots, s_1$, 则它们的状态编码应符合表 5-1 的规定。而且, 这 10 个状态都是必不可少的, 不需要进行状态简化。

选择 JK 触发器组成异步十进制计数器电路。为了便于选取各个触发器的时钟信号, 可以由状态转换图画出电路的时序图, 如图 5-2 所示。

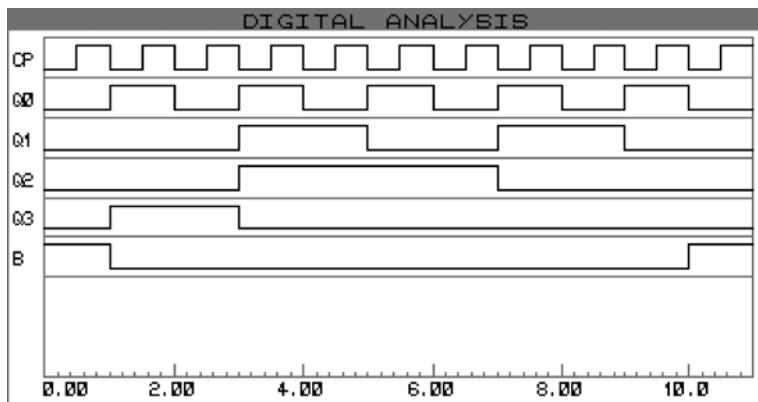


图 5-2 电路的时序图

为触发器选择时钟信号的原则是: 第一, 触发器的状态应该翻转时必须有时钟信号发生; 第二, 触发器的状态不应翻转时“多余的”时钟信号越少越好, 这将有利于触发器状态方程和驱动方程的简化。根据上述原则, 选定 FF_0 的时钟信号 CP_0 为计数输入脉冲, FF_1 的时钟信号 CP_1 取自 \bar{Q}_0 , FF_2 的时钟信号 CP_2 取自 \bar{Q}_1 , FF_3 的时钟信号 CP_3 取自 \bar{Q}_0 。

电路的次态卡诺图如图 5-3 所示。

$Q_1^n Q_0^n$	00	01	11	10
$Q_3^n Q_2^n$	00	0000	0010	0001
01	0011	0100	0110	0101
11	xxxx	xxxx	xxxx	xxxx
10	0111	1000	xxxx	xxxx

图 5-3 次态卡诺图

将图 5-3 进行分解, 可得到 Q_3^{n+1} 、 Q_2^{n+1} 、 Q_1^{n+1} 和 Q_0^{n+1} 的卡诺图, 如图 5-4 所示。

$Q_1^n Q_0^n \backslash Q_3^n Q_2^n$	00	01	11	10
00	1	χ	χ	0
01	0	χ	χ	0
11	χ	χ	χ	χ
10	0	χ	χ	χ

(a) Q_3^{n+1}

$Q_1^n Q_0^n \backslash Q_3^n Q_2^n$	00	01	11	10
00	χ	χ	χ	χ
01	0	χ	χ	χ
11	χ	χ	χ	χ
10	1	χ	χ	χ

(b) Q_2^{n+1}

$Q_1^n Q_0^n \backslash Q_3^n Q_2^n$	00	01	11	10
00	0	χ	χ	0
01	1	χ	χ	0
11	χ	χ	χ	χ
10	1	χ	χ	χ

(c) Q_1^{n+1}

$Q_1^n Q_0^n \backslash Q_3^n Q_2^n$	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	χ	χ	χ	χ
10	1	0	χ	χ

(d) Q_0^{n+1}

图 5-4 卡诺图

在这四个卡诺图中,把没有时钟信号的次态作为任意项来处理,以利于状态方程的化简。另外,由于正常工作时不会出现 $Q_3Q_2Q_1Q_0=1010\sim 1111$ 这 6 个状态,所以也把它们作为卡诺图中的任意项处理。

由图 5-4 所示的卡诺图可得到电路的状态方程为

$$\begin{cases} Q_3^{n+1} = \bar{Q}_3\bar{Q}_2\bar{Q}_1 \cdot cp_3 \\ Q_2^{n+1} = \bar{Q}_2 \cdot cp_2 \\ Q_1^{n+1} = (Q_3 + Q_2\bar{Q}_1) \cdot cp_1 \\ Q_0^{n+1} = \bar{Q}_0 \cdot cp_0 \end{cases} \quad (5-1)$$

式中用小写的 cp_0 、 cp_1 、 cp_2 和 cp_3 表示只有当这些时钟信号到达时,状态方程才是有效的,否则触发器将保持原来的状态不变。 cp_0 、 cp_1 、 cp_2 和 cp_3 在这里只代表 4 个脉冲信号,而不是 4 个逻辑变量。

将式 (5-1) 化为 JK 触发器的标准形式得到

$$\begin{cases} Q_3^{n+1} = [(\bar{Q}_2\bar{Q}_1)\bar{Q}_3 + \bar{1} \cdot Q_3] \cdot cp_3 \\ Q_2^{n+1} = (1 \cdot \bar{Q}_2 + \bar{1} \cdot Q_2) \cdot cp_2 \\ Q_1^{n+1} = [Q_3(Q_1 + \bar{Q}_1) + Q_2\bar{Q}_1] \cdot cp_1 \\ \quad = [(Q_3 + Q_2)\bar{Q}_1 + Q_3 \cdot Q_1] \cdot cp_1 = [(\bar{Q}_3\bar{Q}_2)\bar{Q}_1 + \bar{1} \cdot Q_1] \cdot cp_1 \\ Q_0^{n+1} = (1 \cdot \bar{Q}_0 + \bar{1} \cdot Q_0) \cdot cp_0 \end{cases} \quad (5-2)$$

因为电路正常工作时不会出现 $Q_3Q_1=1$ 的情况,所以在 Q_1^{n+1} 的方程式中删去了这一项。

从式 (5-2) 得到每个触发器应有的驱动方程为

$$\begin{cases} J_3 = \bar{Q}_2\bar{Q}_1 & K_3 = 1 \\ J_2 = K_2 = 1 \\ J_1 = \bar{Q}_3\bar{Q}_2 & K_1 = 1 \\ J_0 = K_0 = 1 \end{cases} \quad (5-3)$$

根据状态转换表画出的输出 B 的卡诺图,如图 5-5 所示。

$Q_3^n Q_2^n \backslash Q_1^n Q_0^n$	00	01	11	10
00	1	0	0	0
01	0	0	0	0
11	χ	χ	χ	χ
10	0	0	χ	χ

图 5-5 输出 B 的卡诺图

由图 5-5 得到

$$B = \bar{Q}_3\bar{Q}_2\bar{Q}_1\bar{Q}_0 \quad (5-4)$$

点选 Component 图标,单击“P”按钮,从弹出的选取元件对话框中选择异步十进制计数器电路仿真元件。仿真元件信息如表 5-2 所示。

表 5-2 仿真元件信息（异步十进制计数器电路分析）

元件名称	所属类	所属子类
74S113 (JK 触发器)	TTL 74S series	Flip-Flops & Latches
7400 (逻辑“与非”门)	TTL 74 series	Gates & Inverters
AND (逻辑“与”门, 2 输入)	Simulator Primitives	Gates
AND_4 (逻辑“与”门, 4 输入)	Modelling Primitives	Digital (Buffer & Gate)

将仿真元件添加到对象选择器后关闭元件选取对话框。

选中对象选择器中的仿真元件，在编辑窗口单击鼠标左键放置仿真元件，按照图 5-6 所示编辑元件并添加连接端子，连接电路。放置数字时钟信号源及电压探针，设置信号源的时钟模式为 Low-High-Low Clock，第一个边沿发生的时刻 First Edge At 为 0，频率 Frequency (Hz) 为 1Hz。

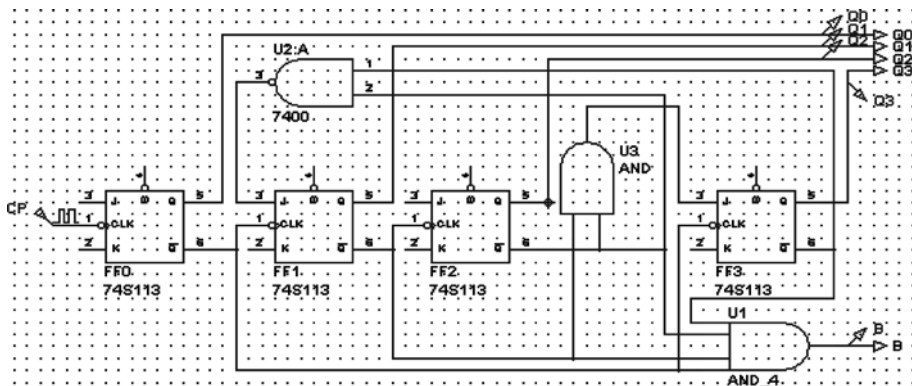


图 5-6 在电路中放置电压探针

5.1.2 异步十进制计数器电路分析

放置数字分析图表。设置图表的仿真开始时间为 0s, 终止仿真时间为 25s。添加信号源与探针，结果如图 5-7 所示。仿真结果如图 5-8 所示。

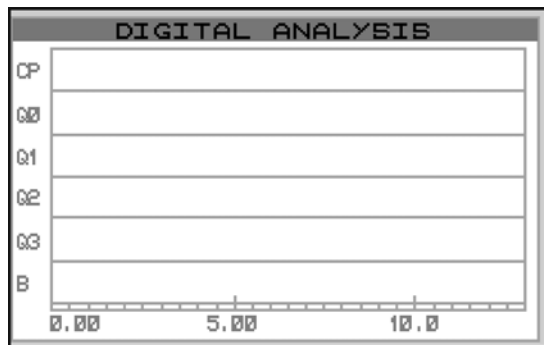


图 5-7 编辑好的数字分析图表

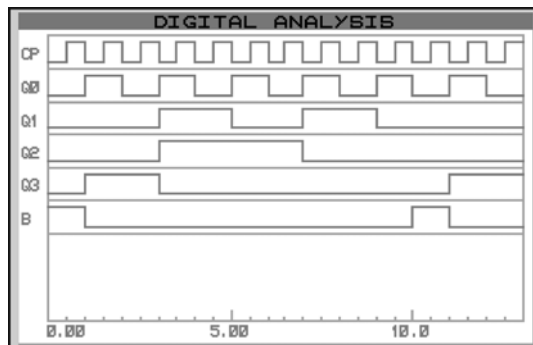


图 5-8 数字分析仿真结果图

从电路的仿真结果可知，系统从 9 开始递减，一直到 0，然后从 9 再次递减，实现了异步十进制计数。

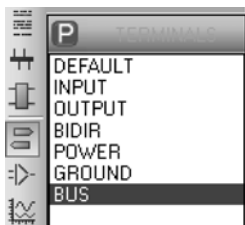


图 5-9 选择总线模式

为了便于查看结果, 现将系统输出以总线形式编辑。单击工具箱中的 BUS 图标, 如图 5-9 所示。

在期望总线起始端出现的位置单击鼠标左键, 然后拖动鼠标将出现总线轮廓, 如图 5-10 所示。

在期望总线路径的拐点处单击鼠标左键, 即可放置拐点。在总线的终点处双击鼠标即可放置总线。

设置鼠标为选择模式, 在单线处单击鼠标左键画线到总线, 如图 5-11 所示。

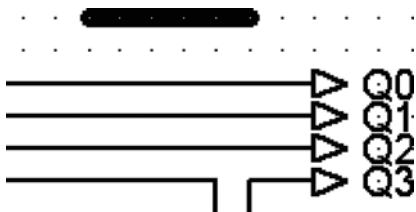


图 5-10 总线轮廓

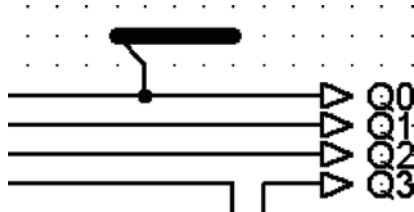


图 5-11 单线与总线连接

按照上述方式连接其他输出端, 如图 5-12 所示。

编辑总线。单击工具箱中的 Wire Label Mode 图标, 如图 5-13 所示。



图 5-12 连接其他输出端到总线

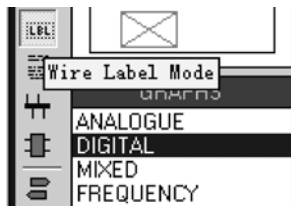


图 5-13 选择 Wire Label Mode 图标

鼠标单击总线, 将弹出如图 5-14 所示的线编辑对话框。在总线上放置电压探针, 如图 5-15 所示。

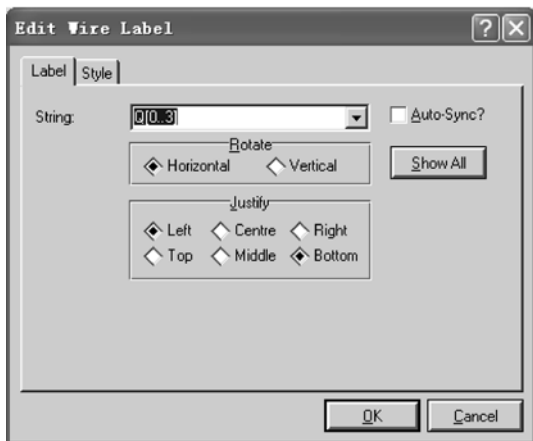


图 5-14 线编辑对话框

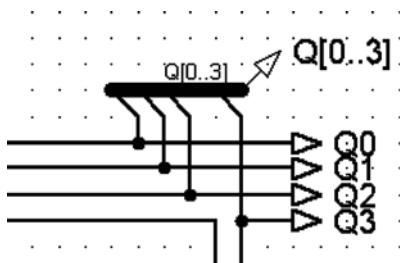


图 5-15 放置电压探针

放置一个新的数字分析图表，添加探针 Q[0..3]、CP 及 B 信号到图表，开始仿真，如图 5-16 所示。

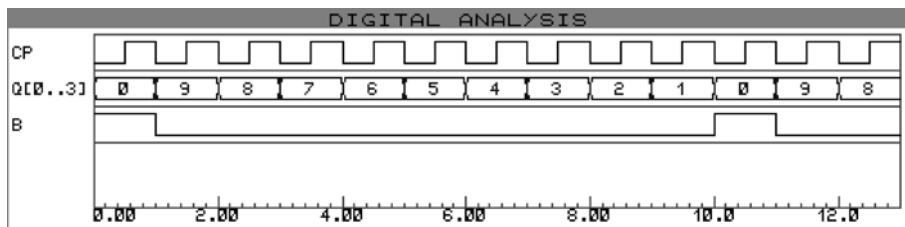


图 5-16 电路仿真结果

从电路的仿真结果可直观地看到系统实现了异步十进制计数。

5.1.3 异步十进制计数器电路完善

从电路的仿真结果可知，电路可实现计数功能，但电路无复位功能。

替换元件。从元件库中取出带有复位端的 JK 触发器 7476，如图 5-17 所示。

从元件外观图可知，7476 元件的 R 引脚为复位端，且为低电平复位。将原电路中的 74S113 替换为 7476。从对象选择器中选择 JK 触发器 7476，在原理图编辑窗口单击鼠标左键，然后将鼠标移到 JK 触发器 74S113 上，如图 5-18 所示。

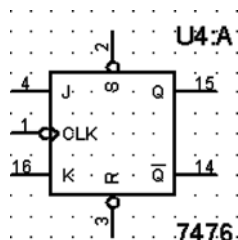


图 5-17 带有复位端的 JK 触发器 7476

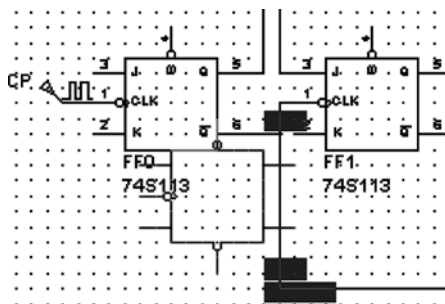


图 5-18 元件替换

确保将要放置的触发器 7476 的轮廓置于将被替换的元件之上，并确保替换元件与被替换元件引脚重合，然后单击鼠标左键替换元件。此时将弹出如图 5-19 所示的对话框。

单击“OK”按钮实现替换。替换后的电路如图 5-20 所示。



图 5-19 是否替换元件对话框

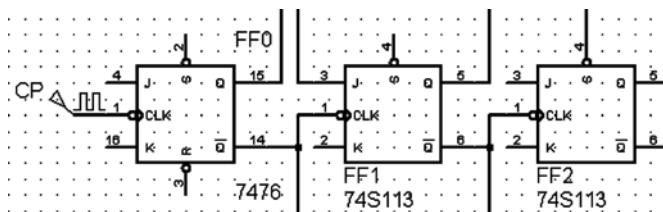


图 5-20 将触发器 74S113 替换为 7476 后的电路

ISIS 在替换元件的同时将保留电路替换前的连线方式。

按上述方式将电路中的其他 74S113 替换为 7476。结果如图 5-21 所示。

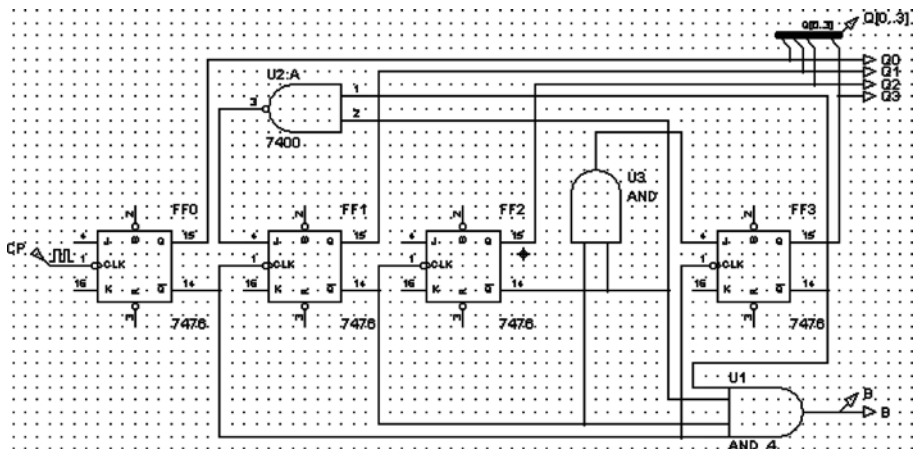


图 5-21 将电路中的其他 74S113 替换为 7476

放置数字单边沿信号源，并将数字单边沿信号源与 JK 触发器的复位引脚相连，设置数字单边沿信号源边沿的极性 Edge Polarity 为 Positive (Low-To-High) Edge，产生边沿的时刻 Edge At (Secs) 为 15s，如图 5-22 所示。

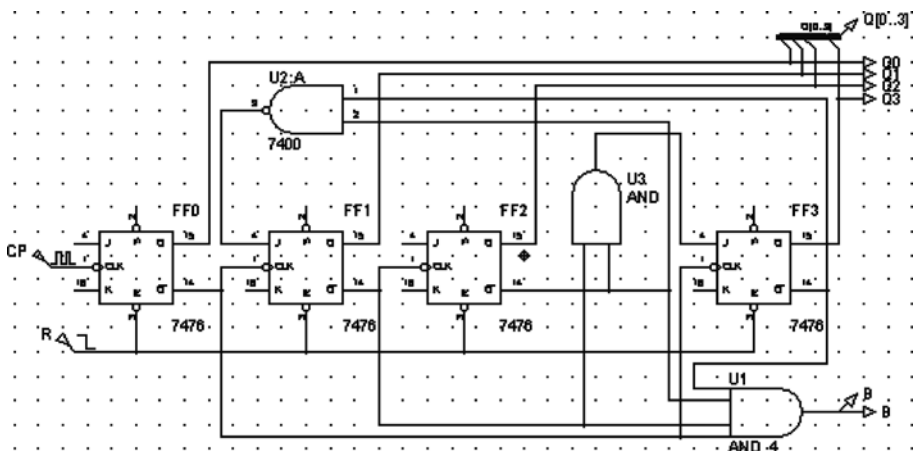


图 5-22 连接数字单边沿信号源与 JK 触发器的复位引脚

电路仿真结果如图 5-23 所示。

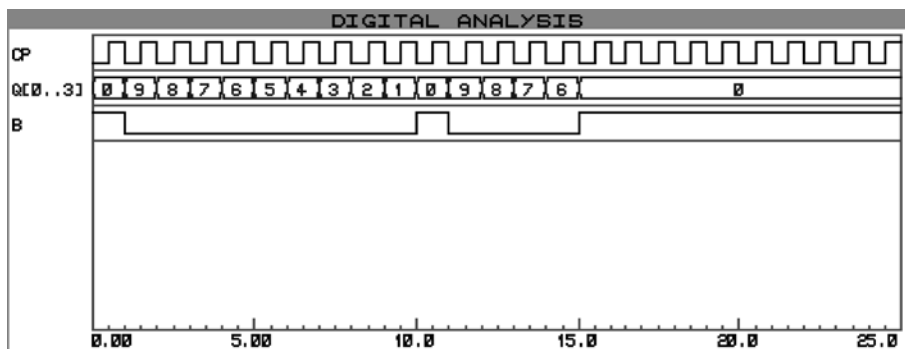


图 5-23 数字分析仿真结果图

从电路的仿真结果可知，当出现复位信号时，系统停止计数，实现复位功能。

将电路中的单边沿信号用数字单周期信号替代。双击单边沿信号源 R，将弹出单边沿信号编辑窗口，点选 Single Pulse，如图 5-24 所示。按照图 5-25 所示编辑数字单周期信号。

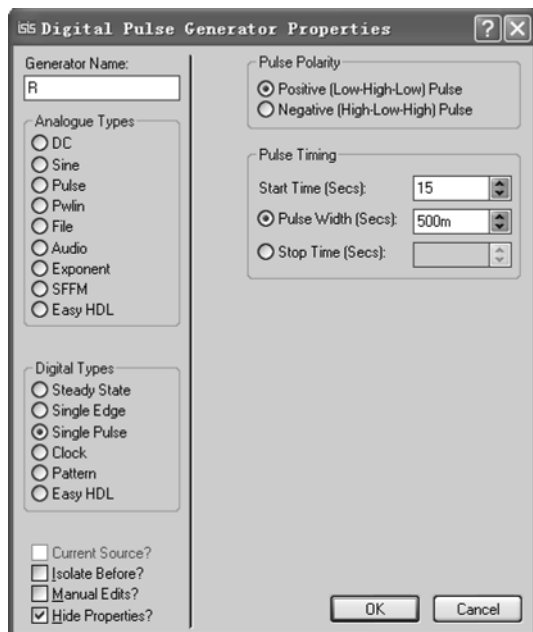


图 5-24 将电路中的单边沿信号用数字单周期信号替代

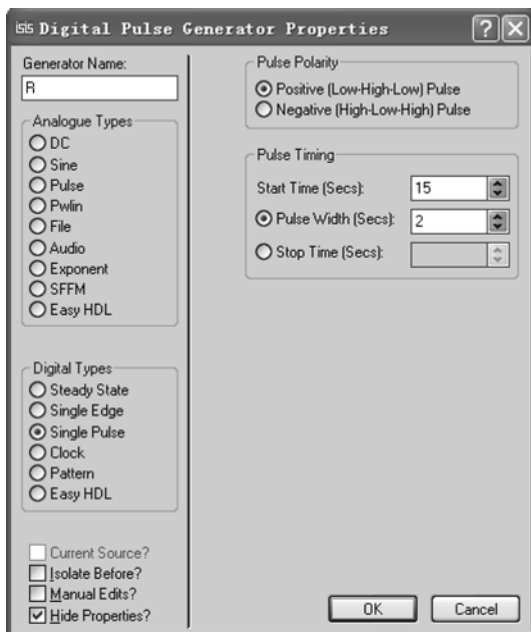


图 5-25 编辑数字单周期信号

编辑完成后单击“OK”按钮完成设置。

仿真电路。电路仿真结果如图 5-26 所示。

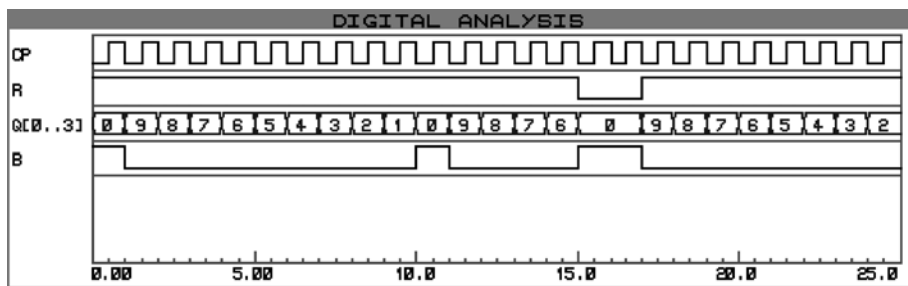


图 5-26 仿真结果图

从电路的仿真结果可知，系统复位后，当复位信号撤销后，重新计数。



5.2 RS 触发器电路分析——数字模式信号源与数字分析

5.2.1 RS 触发器电路

【基本 RS 触发器】把两个“与非”门的输入、输出端交叉连接，即可构成基本 RS 触发器，如图 5-27 所示。

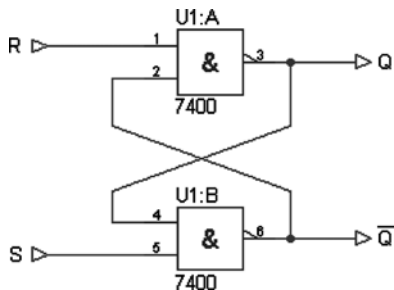


图 5-27 两个“与非”门构成的基本 RS 触发器

基本 RS 触发器有两个输入端 R、S 和两个输出端 Q、 \bar{Q} 。根据与非门的逻辑关系，触发器的逻辑表达式为： $Q = S\bar{Q}$ ， $\bar{Q} = RQ$ 。

根据输入信号 R、S 不同状态的组合，触发器的输出与输入之间的关系如表 5-3 所示。

表 5-3 用两个与非门组成的基本 RS 触发器的功能表

R	S	Q
1	0	1
0	1	0
1	1	不变
0	0	不定

點選 Component 图标，单击“P”按钮，从弹出的选取元件对话框中选择 RS 触发器电路仿真元件。仿真元件信息如表 5-4 所示。

表 5-4 仿真元件信息（RS 触发器电路分析）

元件名称	所属类	所属子类
NAND_2 (“与非”门)	TTL 74S series	Flip-Flops & Latches

将仿真元件添加到对象选择器后关闭元件选取对话框。

选中对象选择器中的仿真元件，在编辑窗口单击鼠标左键放置仿真元件，如图 5-28 所示。

连接电路。在 NAND_2 的连接端子单击鼠标开始画线，在期望的拐点击鼠标左键，即可放置拐点，如图 5-29 所示。在期望绘制斜线的部分，按 Ctrl 键即可绘制斜线，如图 5-30 所示。

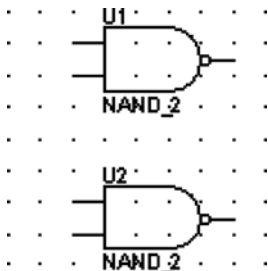


图 5-28 放置仿真元件

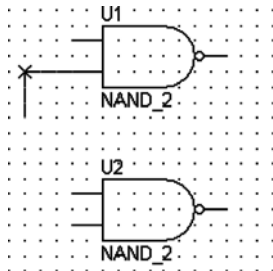


图 5-29 绘制连线拐点

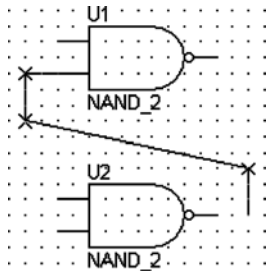


图 5-30 绘制斜线

按上述方式连接电路。结果如图 5-31 所示。按图 5-32 所示编辑端口。

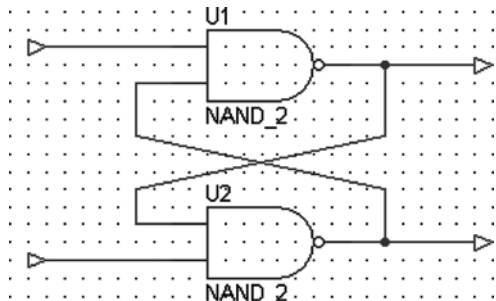


图 5-31 RS 触发器电路

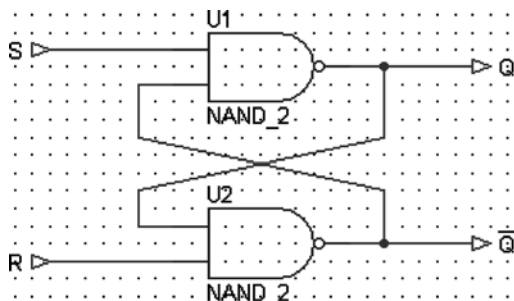


图 5-32 编辑好的 RS 触发电路

5.2.2 RS 触发器电路调试

放置调试工具。点选 Component 图标，单击“P”按钮，将弹出元件选取对话框。从元件库中选取数字电路调试工具。调试工具信息如表 5-5 所示。

表 5-5 调试工具信息（RS 触发器电路调试）

元件名称	所属类	所属子类
LOGICSTATE（逻辑状态）	Debugging Tools	Logic Stimuli
LOGICPROBE(BIG)（逻辑探针）	Debugging Tools	Logic Probes

将“逻辑状态”调试元件放置到电路。在编辑窗口单击鼠标左键，即可放置“逻辑状态”调试元件到电路，并将元件与电路连接，如图 5-33 所示。

按上述方式放置多个“逻辑状态”调试元件及“逻辑探针”调试元件，并与电路的输入、输出端口连接。结果如图 5-34 所示。

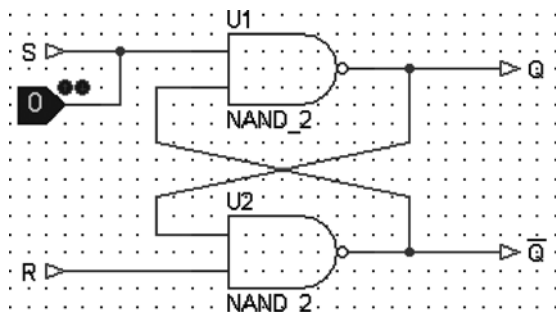


图 5-33 放置“逻辑状态”调试元件到电路

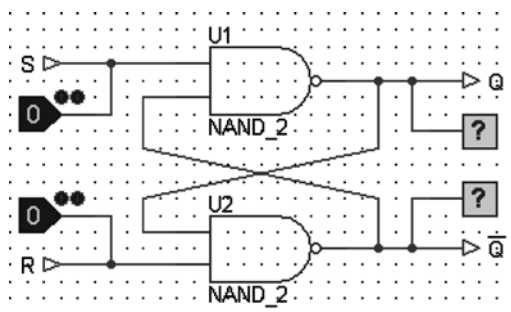


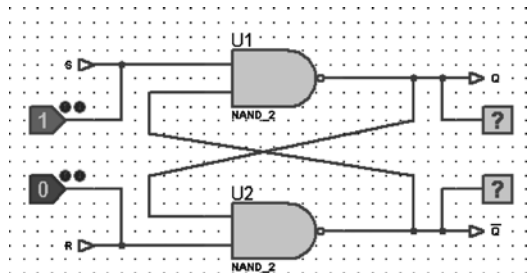
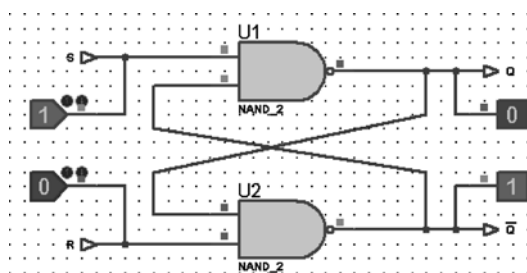


图 5-34 放置多个“逻辑状态”调试元件及“逻辑探针”调试元件到电路

调试电路。将电路的输入信号设置为：S=1，R=0。单击“逻辑状态”调试元件的活性标识  即可实现信号的设置，如图 5-35 所示。

仿真电路。单击控制面板中的“运行”按钮 ，电路开始仿真。仿真结果如图 5-36 所示。

图 5-35 设置电路输入信号为: $S=1, R=0$ 图 5-36 $S=1, R=0$ 时电路的仿真结果

从电路的调试仿真结果可知, 当输入信号为: $S=1, R=0$ 时, 系统输出为: $Q=0, \bar{Q}=1$, 符合表 5-5 列出的功能。即由两个“与非”门组成的 RS 触发器的 R 端称为置 0 端, 且为低电平有效。

分别设置输入信号为: $S=0, R=1$; $S=1, R=1$; $S=0, R=0$ 调试电路。电路的仿真结果如图 5-37 所示。

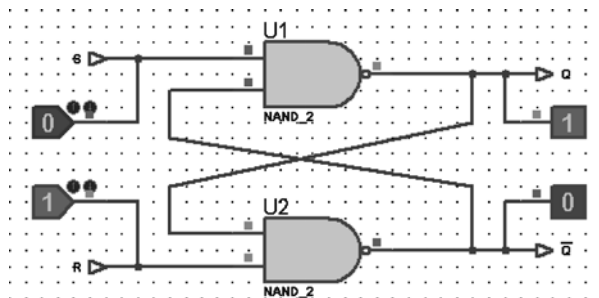
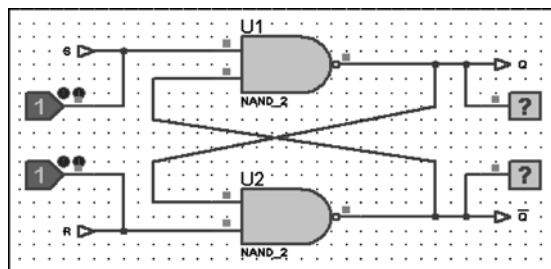
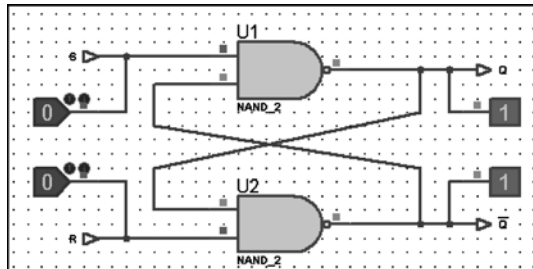
(a) $S=0, R=1$ 时电路的仿真结果(b) $S=1, R=1$ 时电路的仿真结果(c) $S=0, R=0$ 时电路的仿真结果

图 5-37 不同输入信号下电路的仿真结果

从上述仿真结果可知, S 为 RS 触发器的置 1 端, 且为低电平有效。而当 $S=1, R=1$ 及 $S=0, R=0$ 时无法确定电路的功能。因此用数字图表仿真电路输出。

5.2.3 RS 触发器数字图表分析——数字模式信号源编辑

在电路中添加直流仿真输入源。单击 Generator 图标, 系统在对象选择窗口列出各种信号源, 点选 DPATTERN 信号源, 在编辑窗口单击鼠标左键, 放置数字模式信号源, 并将数字模式信号源与 RS 触发器的 R、S 引脚相连, 如图 5-38 所示。

双击数字模式信号源, 将弹出如图 5-39 所示的数字模式信号源编辑对话框。

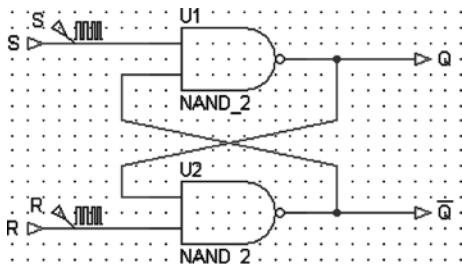


图 5-38 连接数字模式信号源与 RS 触发器的 R、S 引脚

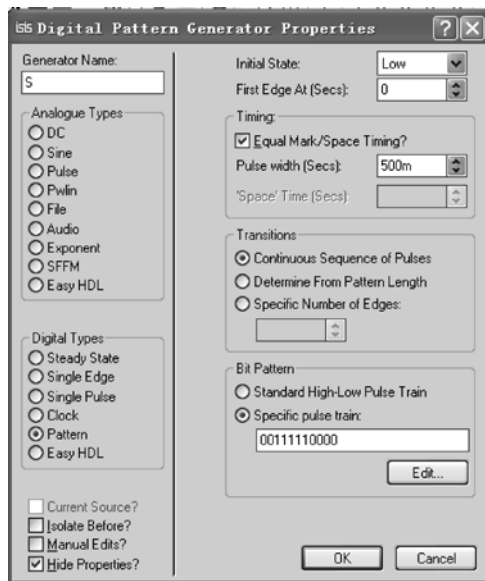


图 5-39 数字模式信号源编辑对话框

也可点选 EDIT，在出现的对话框中用左键来绘制波形，如图 5-40 所示。

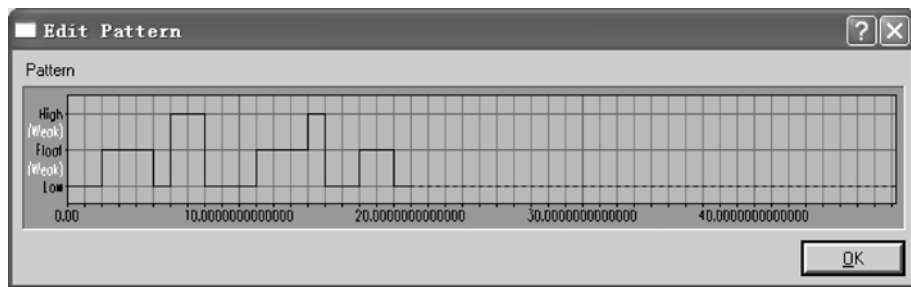


图 5-40 DPattern 脉冲序列编辑对话框

在编辑模式脉冲序列时，可使用如下字符：

- ☺ 0 或 L-强逻辑 0。
- ☺ 1 或 H-强逻辑 1。
- ☺ l-弱逻辑 0。
- ☺ h-弱逻辑 1。
- ☺ F 或 f-浮动电平。

按图 5-39 所示编辑信号源。

参照上述方式编辑 R 信号，如图 5-41 所示。

5.2.4 RS 触发器数字图表分析——探针及数字分析图表编辑

放置测量探针。使用旋转或镜像按钮调整探针的方向后，在编辑窗口期望放置探针的位置单击鼠标左键，电压探针被放置到电路图中，如图 5-42 所示。

放置数字分析图表。将信号源及测量探针添加到数字分析图表，设置图表的终止仿真时间为 6s。结果如图 5-43 所示。

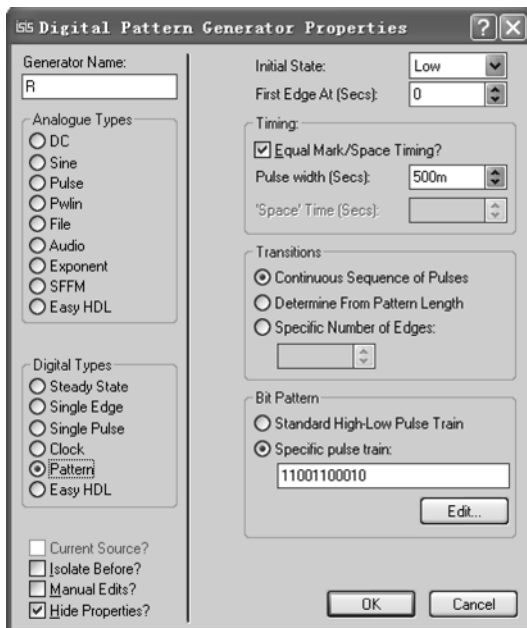


图 5-41 编辑 R 信号

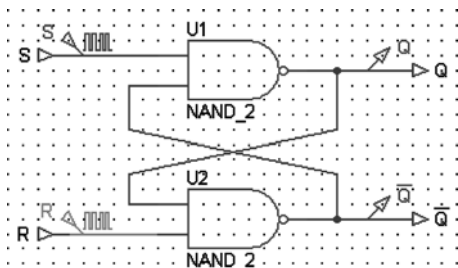


图 5-42 添加电压探针

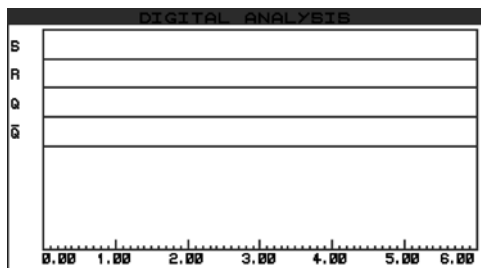


图 5-43 编辑好的数字分析图表

5.2.5 RS 触发器电路分析

电路仿真结果如图 5-44 所示。

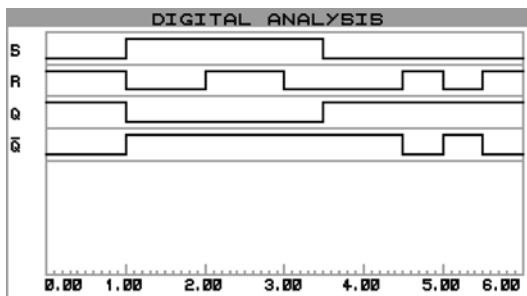
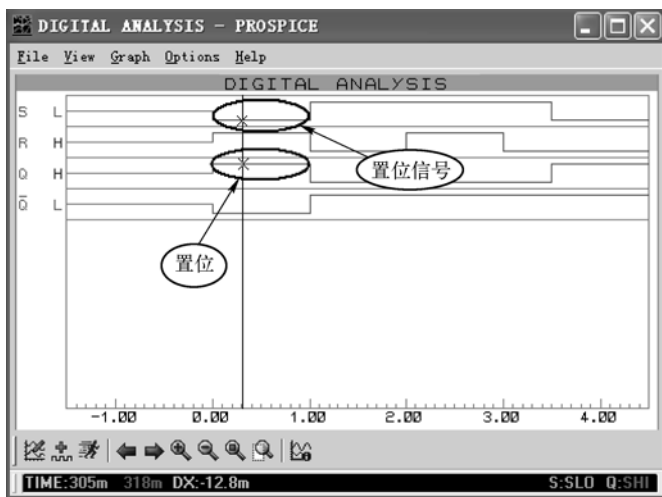
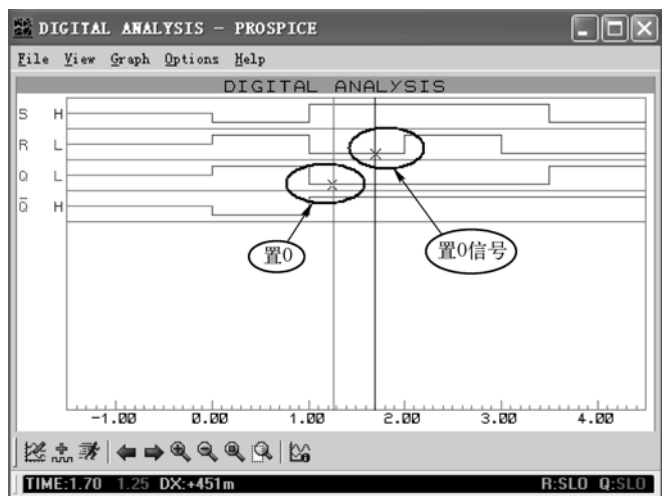


图 5-44 数字分析仿真结果图

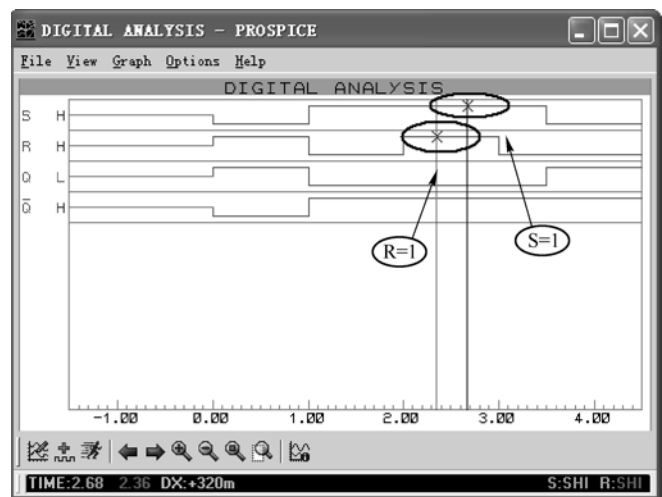
单击图表表头，图表将以窗口形式出现。在窗口单击鼠标左键放置测量探针，测量 R、S 值与 Q、 \bar{Q} 的关系，如图 5-45 所示。



(a) S 作为置 1 信号时系统的输出



(b) R 作为置 0 信号时系统的输出



(c) R=1, S=1 时系统保持原来状态

图 5-45 结果分析 (R、S 值与 Q、 \bar{Q} 的关系)

5.2.6 RS 触发器用于消除机械开关振荡引起的脉冲

【机械开关振荡引起的脉冲】机械开关接通时，用于振动会使电压或电流波形产生“毛刺”，如图 5-46 所示。

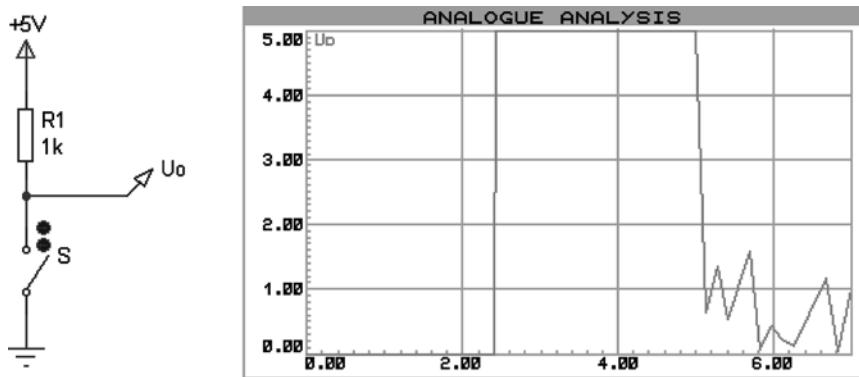


图 5-46 机械开关及其输出电压波形

在电子电路中，一般不允许出现这种现象，因为这种干扰信号会导致电路工作出错。利用 RS 触发器电路的记忆作用可以消除上述开关振荡所产生的影响。

利用 RS 触发器消除机械开关振动的电路如图 5-47 所示。仿真元件信息如表 5-6 所示。

表 5-6 仿真元件信息 (RS 触发器消除机械开关振动)

元件名称	所属类	所属子类
NAND_2 (“与非”门)	TTL 74S series	Flip-Flops & Latches
RES (电阻)	Resistors	Generic
SW-SPDT (单刀双掷开关)	Switches & Relays	Switches

仿真电路。在本电路的仿真中使用分段线性信号源作为仿真电路的输入信号。仿真电路如图 5-48 所示。

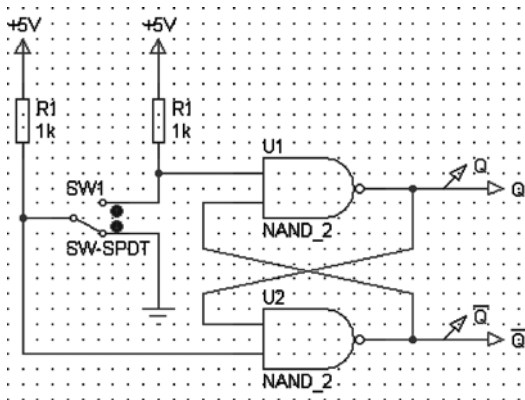


图 5-47 RS 触发器消除机械开关振动电路

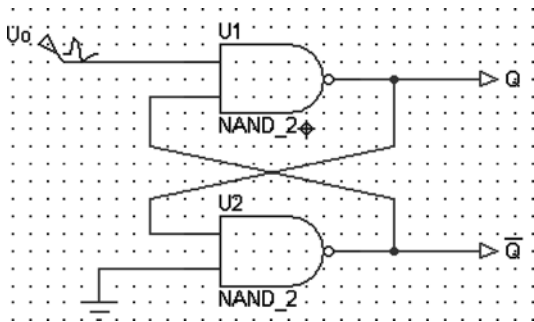
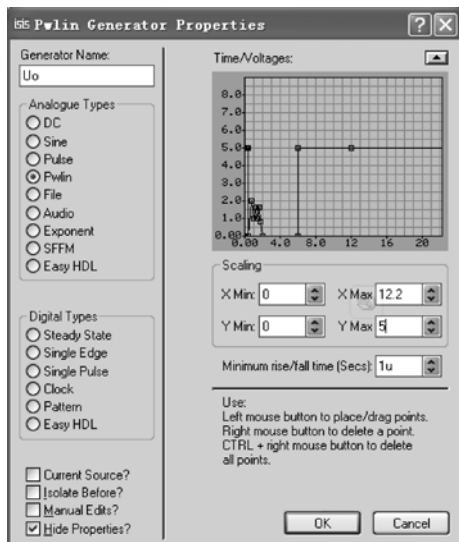
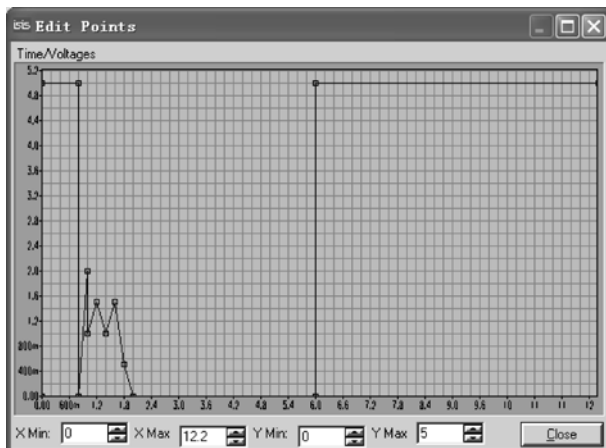


图 5-48 RS 触发器消除机械开关振动仿真电路

其中信号源的设置如图 5-49 所示。



(a) 分段线性信号源的设置



(b) 设置信号源节点位置

图 5-49 分段线性信号源及节点设置

在电路中添加测量探针，结果如图 5-50 所示。

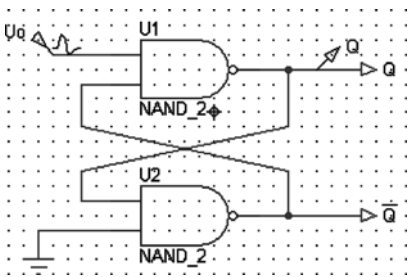


图 5-50 在电路中添加测量探针

放置混合仿真图表。选择 Graph→Add Trace 菜单命令，将弹出如图 5-51 所示的对话框。

按照图 5-50 所示编辑“添加瞬态曲线”对话框，编辑完成后单击“OK”按钮，此时模拟分段线性信号源被添加到混合分析图表中。再次选择 Graph→Add Trace 菜单命令，按图 5-52 所示编辑对话框。

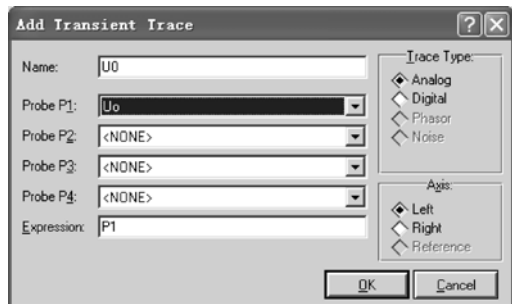


图 5-51 添加曲线

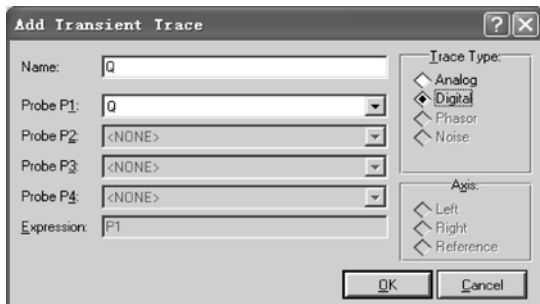


图 5-52 将输出信号作为数字信号添加到图表

此时混合图表中的信号如图 5-53 所示。设置混合分析图表的终止仿真时间为 10s。电路仿真结果如图 5-54 所示。

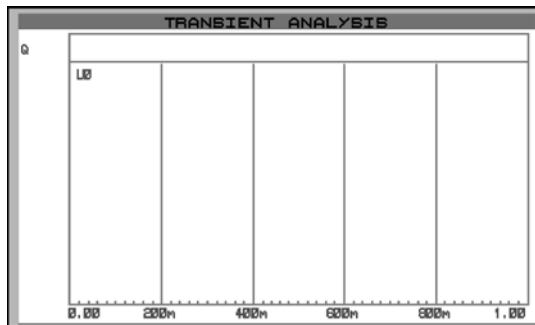


图 5-53 混合图表中的信号

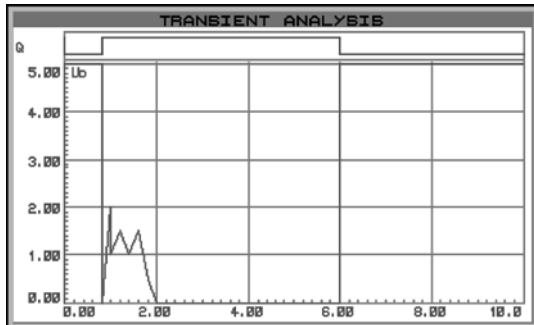


图 5-54 消除振动电路仿真结果

从系统的仿真结果可知，Q 端输出的信号没有“毛刺”。



5.3 竞赛抢答器电路分析——数字单周期脉冲信号源与数字分析

【竞赛抢答器】以四人抢答电路为例。四人参加比赛，每人一个按钮，其中一人按下按钮后，相应的指示灯点亮，并且其他之后按下的按钮不起作用。要求以 74LS171 四 D 触发器为核心器件设计四人竞赛抢答电路。

74LS171 内部包含了四个 D 触发器，各输入、输出以序号相区别，引脚如图 5-55 所示。

以 74LS171 四 D 触发器为核心器件设计的四人竞赛抢答器电路如图 5-56 所示。

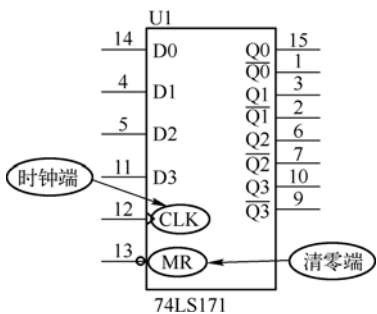


图 5-55 74LS171 引脚图

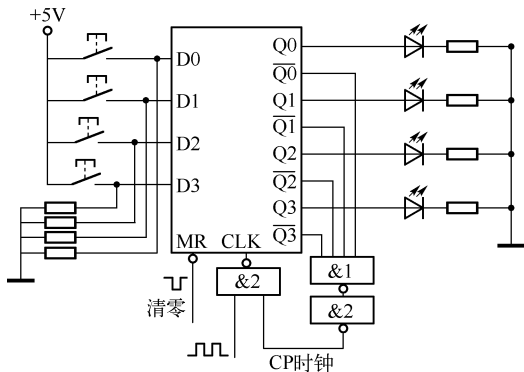


图 5-56 74LS171 四 D 触发器为核心器件设计的四人竞赛抢答器电路

其中清零信号用于赛前清零，清零后电路结果如图 5-57 所示。此时四个发光二极管均熄灭，电路的反相端输出均为 1，时钟端“与”门开启，等待输入信号。

当第一个按钮被按下时，Q0 端输出信号为高，点亮发光二极管，而 $\overline{Q0}$ 端输出信号为低，如图 5-58 所示。

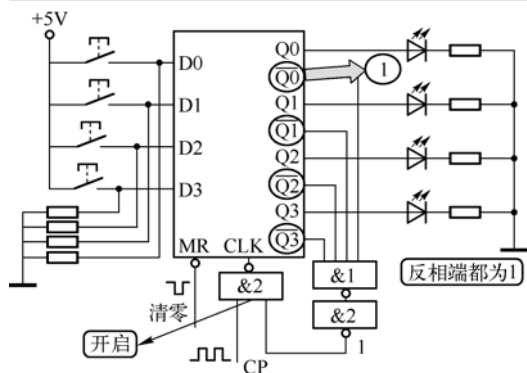
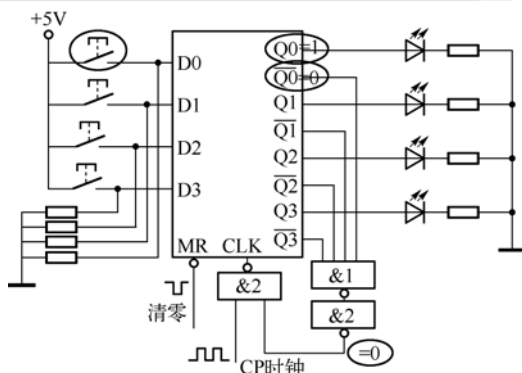


图 5-57 电路清零

图 5-58 当第一个按钮被按下时, Q_0 端、 $\overline{Q_0}$ 端的输出

当 $\overline{Q_0}$ 端输出信号为低时, 74LS171 时钟端被封, 此后其他输入信号对系统输出不起作用。

5.3.1 竞赛抢答器电路

点选 Component 图标, 单击“P”按钮, 从弹出的选取元件对话框中选择竞赛抢答器电路仿真元件。仿真元件信息如表 5-7 所示。

表 5-7 仿真元件信息 (竞赛抢答器电路分析)

元件名称	所属类	所属子类
74LS171 (四 D 触发器)	TTL 74S series	Flip-Flops & Latches
74LS20 (四输入与门)	TTL 74S series	Gates & Inverters
74LS00 (二输入与门)	TTL 74S series	Gates & Inverters
RES (电阻)	Resistors	Generic
BUTTON (按钮)	Switches & Relays	Switches
LED-GREEN (绿色指示灯)	Optoelectronics	LEDs

将仿真元件添加到对象选择器后关闭元件选取对话框。

选中对象选择器中的仿真元件, 在编辑窗口单击鼠标左键放置仿真元件, 按图 5-59 所示编辑元件, 添加连接端子, 并连接电路。

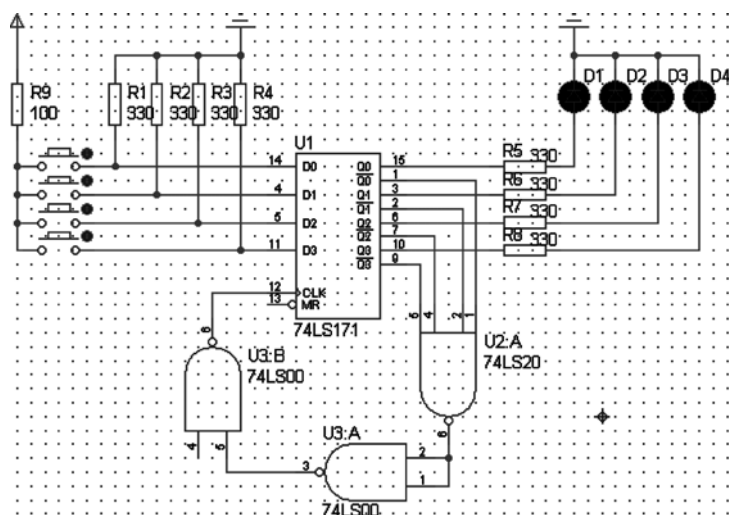


图 5-59 竞赛抢答器电路 (含参数)



图 5-60 点选文本编辑图标

标注设计。使用文本编辑 (Text Scripts) 标注电路。单击工具箱中的 Text Script Mode 图标, 如图 5-60 所示。

在期望放置标注的位置单击鼠标左键, 将出现如图 5-61 所示的 Edit Script Block 对话框。

在 Text 区域输入如图 5-61 所示文本后, 单击“OK”按钮, 完成编辑。结果如图 5-62 所示。

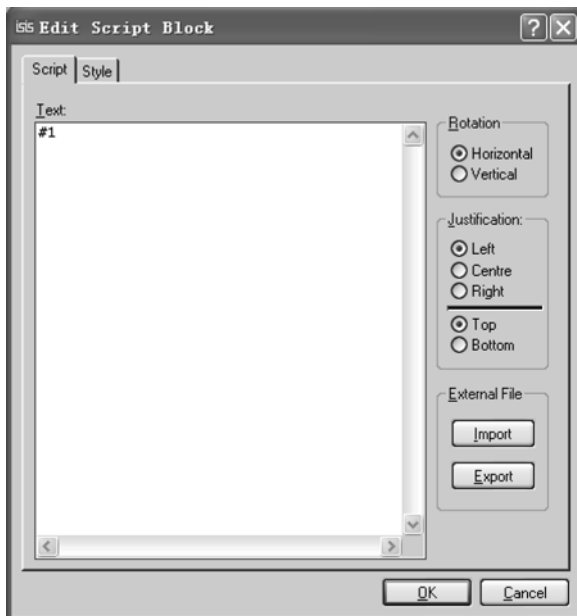


图 5-61 Edit Script Block 对话框

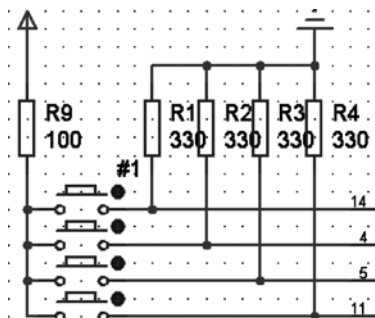


图 5-62 标注按钮

按照上述方式编辑标注其他按钮, 结果如图 5-63 所示。

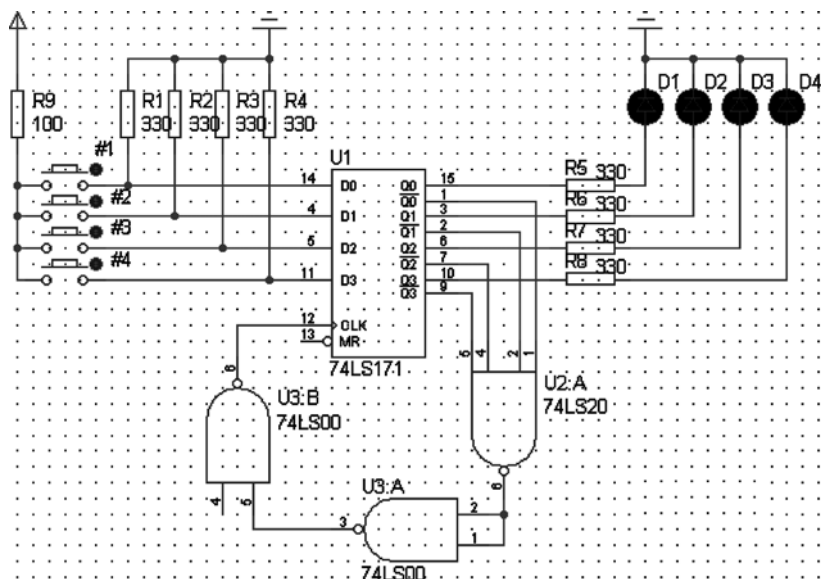


图 5-63 标注其他按钮

5.3.2 数字时钟信号源及数字单周期脉冲信号源编辑

在电路中添加数字时钟仿真输入源, 设置数字时钟信号源的频率为 1kHz, 时钟模式 Clock Type 为 Low-High-Low Clock, 如图 5-64 所示。

放置数字单周期脉冲信号源, 并将数字单周期脉冲信号源与 74LS171 的清零引脚相连, 设置单周期脉冲信号源的脉冲极性 Pulse Polarity 为负脉冲 Negative Pulse, 起始时刻 Start Time (Secs) 为 0s, 脉宽 Pulse Width (Secs) 为 500ms, 如图 5-65 所示。

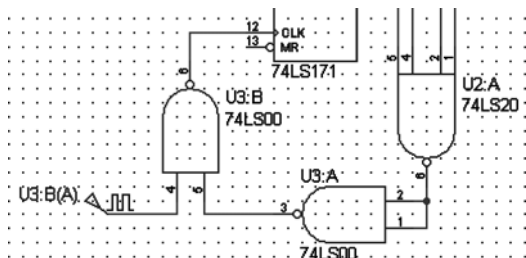


图 5-64 连接数字时钟信号源
与 U3: B 的输入引脚

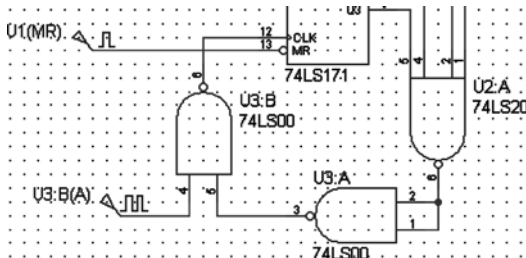


图 5-65 连接数字单周期脉冲信号源
与 74LS171 的清零引脚

编辑好的竞赛抢答器仿真电路如图 5-66 所示。

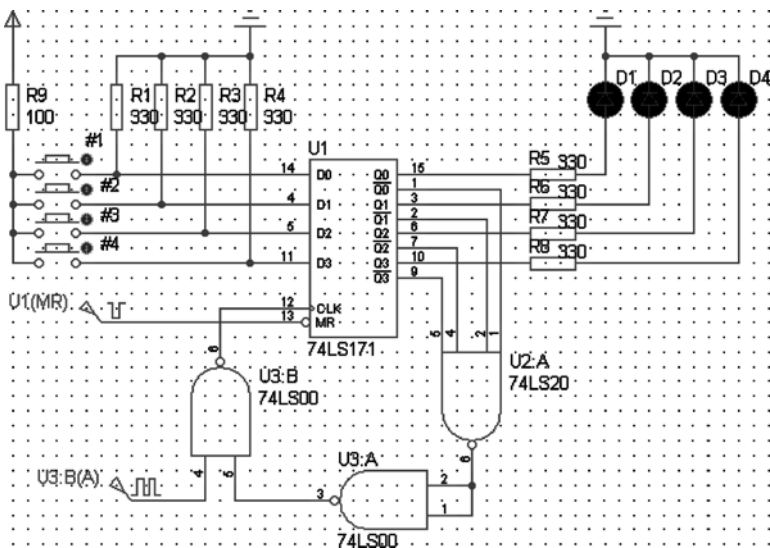


图 5-66 竞赛抢答器仿真电路

5.3.3 竞赛抢答器电路分析

单击控制面板中的运行按钮, 系统进入仿真状态, 如图 5-67 所示。

从系统的仿真图可知, 系统经清零后, LED 全部熄灭, 且系统输入时钟有效。

(1) 当按下#1 键后, 系统的仿真结果如图 5-68 所示。

从系统的仿真结果可知, 按下#1 键后, D1 发光二极管点亮, 同时系统的时钟输入端被锁定。在上述情形下, 按动其他按键, 系统不响应动作, 如图 5-69 所示。

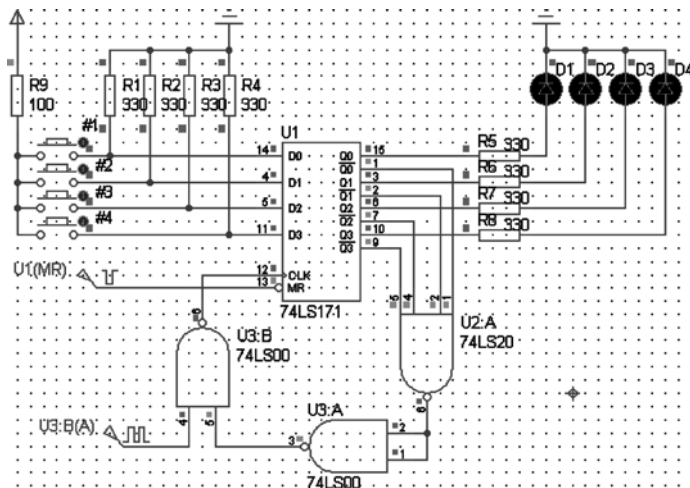


图 5-67 系统进入仿真状态

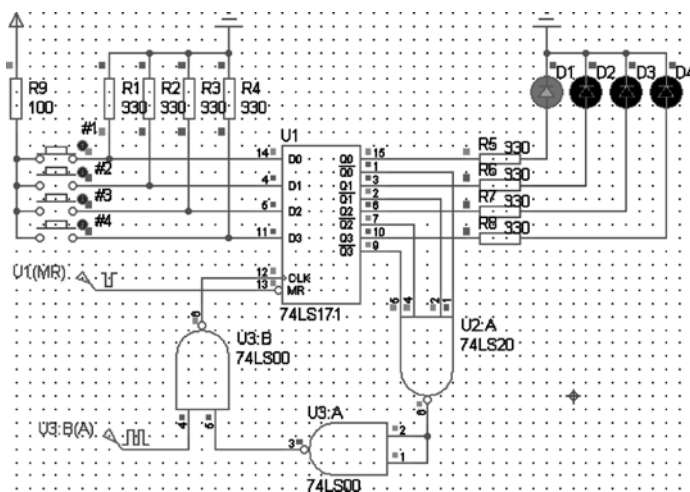
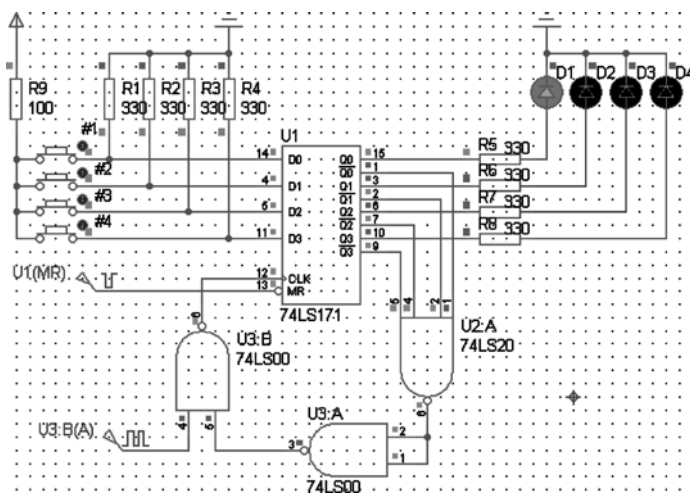
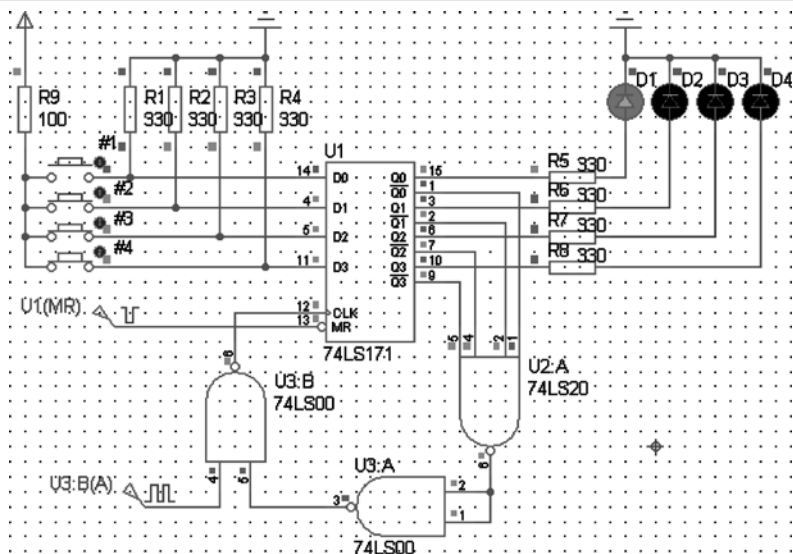


图 5-68 按下#1 键后系统的仿真结果



(a) #1 键释放后, 系统仍不响应其他按键的操作

图 5-69 系统不响应其他按键的操作



(b) #1 键按下状态, 系统不响应其他按键的操作

图 5-69 系统不响应其他按键的操作 (续)

(2) 改变限流电阻, 观察指示灯的变化。限流电阻的作用是减小负载端电流, 在发光二极管一端添加一个限流电阻可以减小流过发光二极管的电流, 防止损坏。

☺ 当限流电阻 R5~R8 的阻值为 1kΩ 时, 电路如图 5-70 所示。

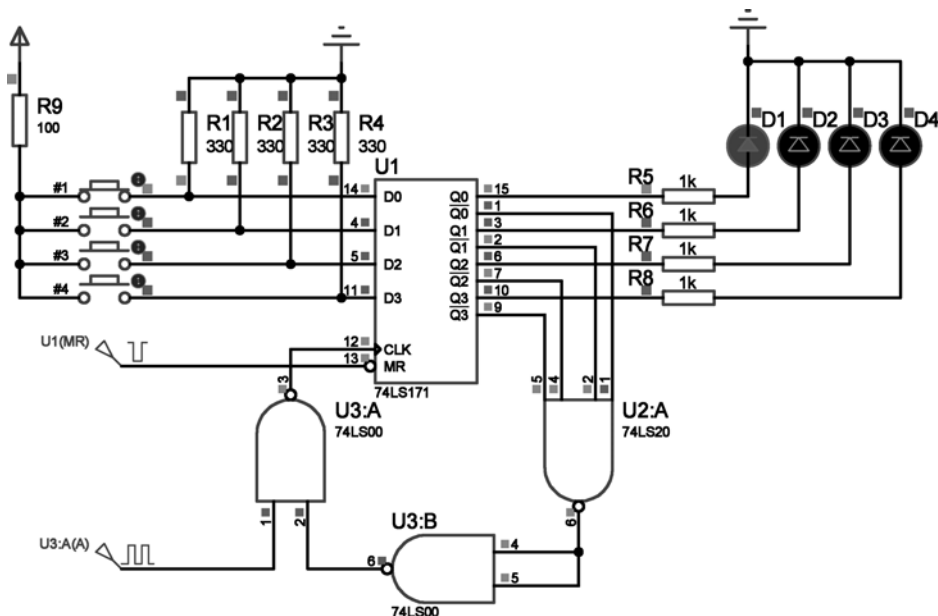
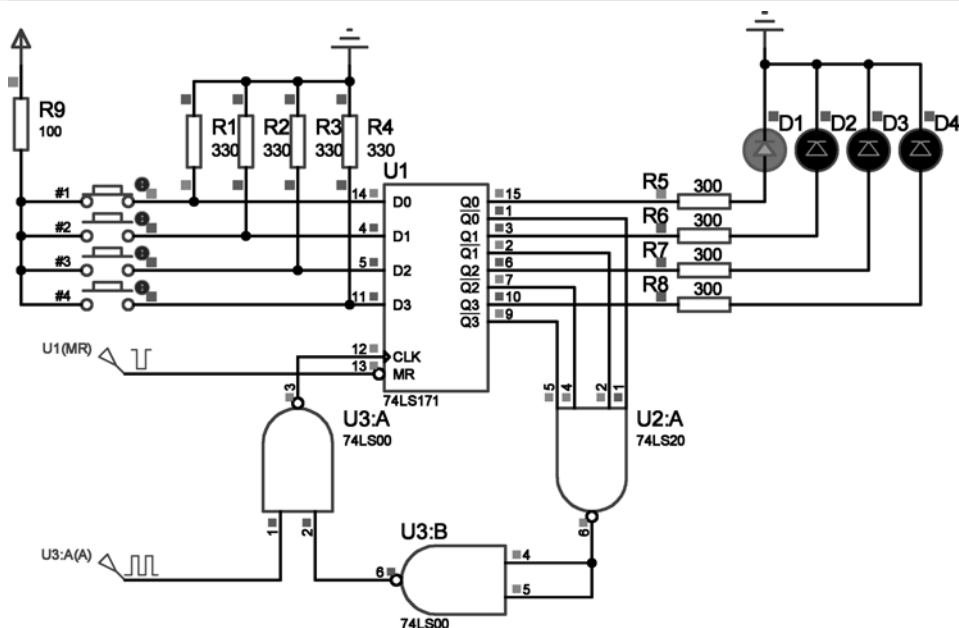


图 5-70 限流电阻阻值为 1kΩ 的仿真结果

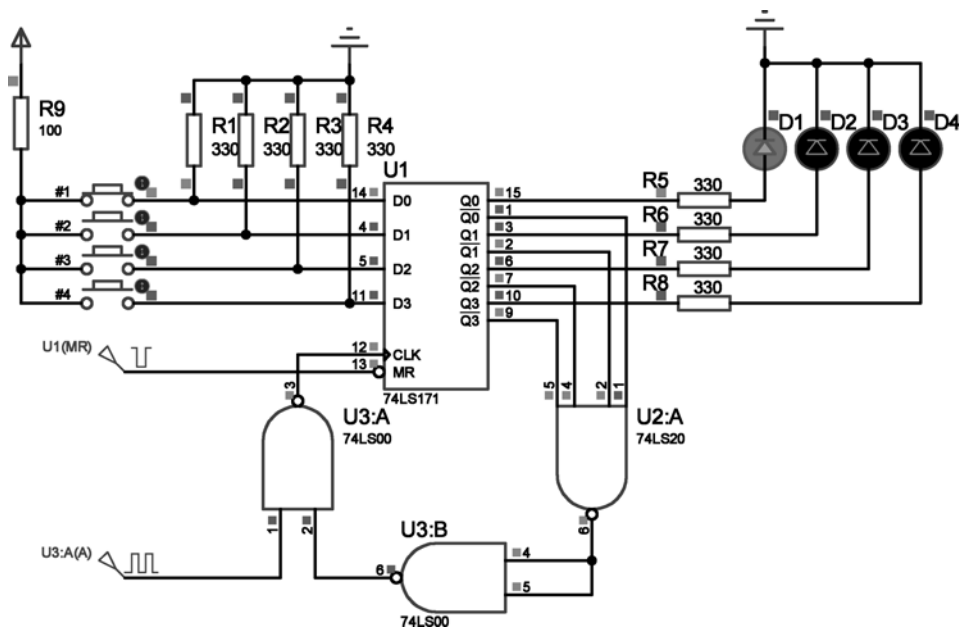
☺ 当限流电阻 R5~R8 的阻值为 300Ω 时, 电路如图 5-71 所示。

由比较可以得出, 限流电阻越小, 指示灯越亮。每种指示灯用的 LED 工作电流均为 10mA 左右, 电流过大就会影响指示灯的寿命。并且因为不同颜色的 LED 端压降不同, 比如蓝光、白光的通常为 3V 左右, 高亮的为 2.5V 左右, 普通亮度的为 1.5~2V, 因此在选择限流电阻的时候要考虑 LED 指示灯所需电流, 电流太大或者太小都会影响指示灯正常工作。

图 5-71 限流电阻阻值为 300Ω 的仿真结果

(3) 改变下拉电阻观察指示灯变化。电阻因为是接地，所以称为下拉电阻，是将电路一端的电平向低方向（地）拉。下拉电阻的主要作用是与上接电阻一起在电路驱动器关闭时给线路（节点）以一个固定的电平。可以加大输出引脚的驱动能力，可以提高输出的高电平值，另外下拉电阻还可以提高抗电磁干扰能力。

☺ 下拉电阻 $R1 \sim R4$ 的阻值为 330Ω 时，电路如图 5-72 所示。

图 5-72 下拉电阻阻值为 330Ω 时的仿真结果

☺ 当下拉电阻 $R1 \sim R4$ 的阻值为 $5k\Omega$ 时，电路如图 5-73 所示。

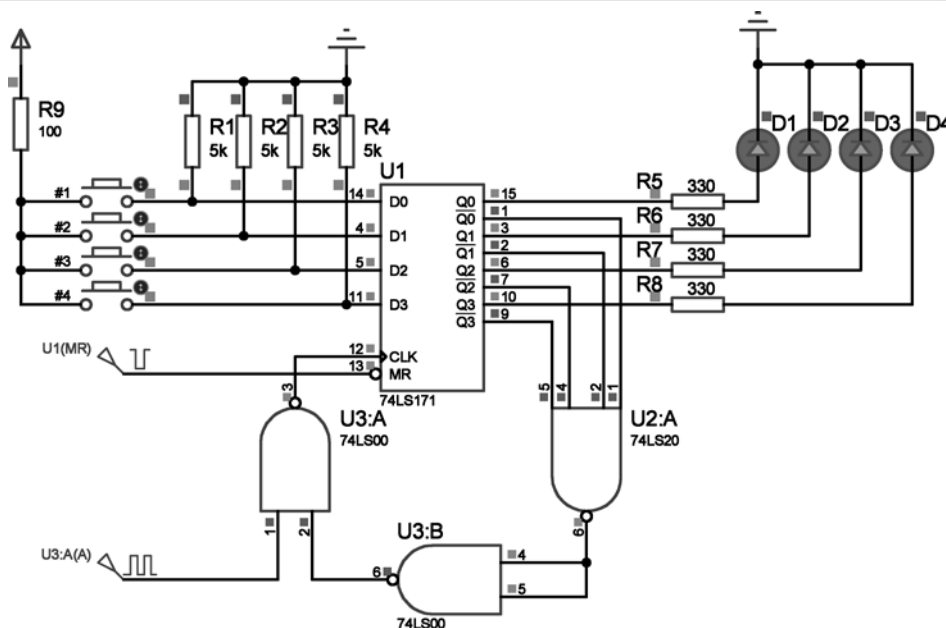


图 5-73 下拉电阻阻值为 5kΩ 时的仿真结果

由于 TTL 门电路的特点是当悬空时为高电平，TTL 电路规定高电平阈值是大于 3.4V，如果要加高电平信号的话，必须要保证输入电压大于 3.4V。通过电路计算理论上当串联大于 1.4kΩ 的电阻时，输入端呈现高电平。因此当输入端串联 5kΩ 电阻后，再输入低电平，输入端呈现高电平，而实际中需要串联小于 2.4kΩ 的电阻，输入的低电平才会被识别。

5.3.4 利用灌电流和或非门设计竞赛抢答器电路

1. 放置仿真元件

点选 Component 图标，单击“P”按钮，从弹出的选取元件对话框中选择竞赛抢答器电路仿真元件。仿真元件信息如表 5-8 所示。

表 5-8 仿真元件信息（竞赛抢答器电路分析）

元 件 名 称	所 属 类	所 属 子 类
74HC175（四 D 触发器）	TTL 74HC series	Flip-Flops & Latches
74HC4002（四输入或非门）	TTL 74HC series	Gates & Inverters
74HC02（二输入或非门）	TTL 74HC series	Gates & Inverters
RES（电阻）	Resistors	Generic
BUTTON（按钮）	Switches & Relays	Switches
LED-GREEN（绿色指示灯）	Optoelectronics	LEDs

74HC 系列是高速集成电路，74LS 系列是低速集成电路。在实际使用的时候可以使用高速集成电路来代替低速集成电路，但不可以使用低速集成电路来代替高速集成电路。

选中对象选择器中的仿真元件，在编辑窗口单击鼠标左键放置仿真元件，并连接电路。结果如图 5-74 所示。

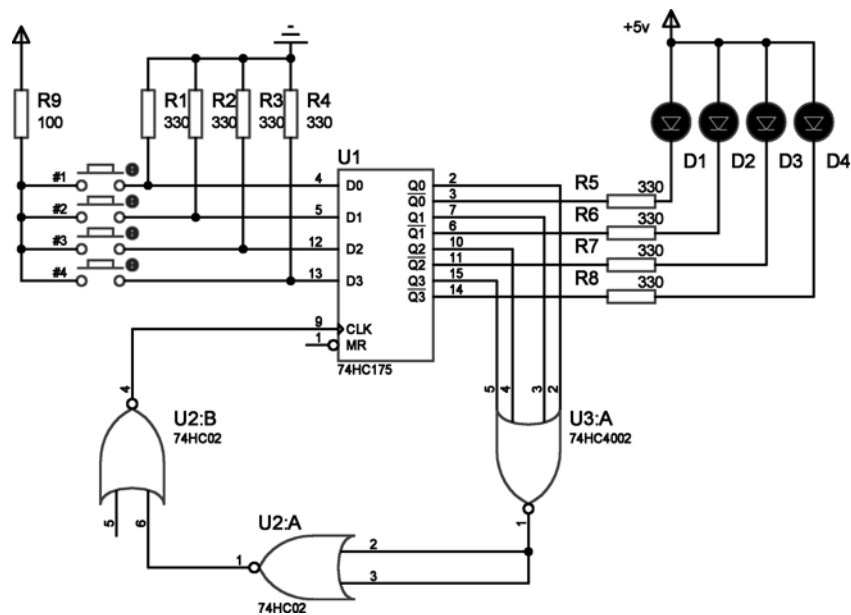


图 5-74 竞赛抢答器电路 (含参数)

2. 添加信号源

与之前的竞赛抢答器输入信号相同, 时钟信号 (CLK) 为下降波, 脉冲宽度为 500ms。
U2: B 输入信号为“高-低-高”类型的脉冲, 频率为 1kHz。

3. 进行仿真

按下运行键后, 进入仿真状态, 如图 5-75 所示。按下#1 键后系统的仿真结果如图 5-76 所示。

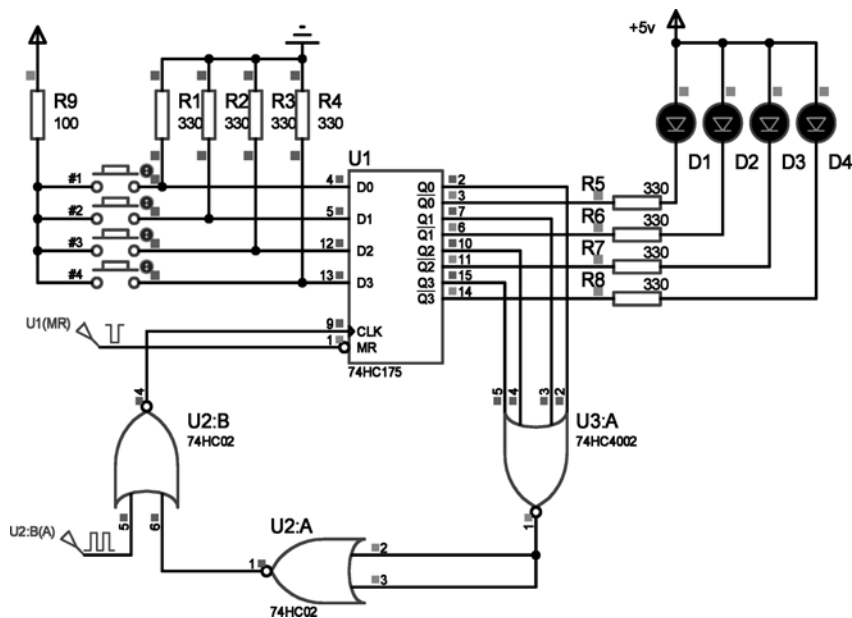


图 5-75 进入仿真状态

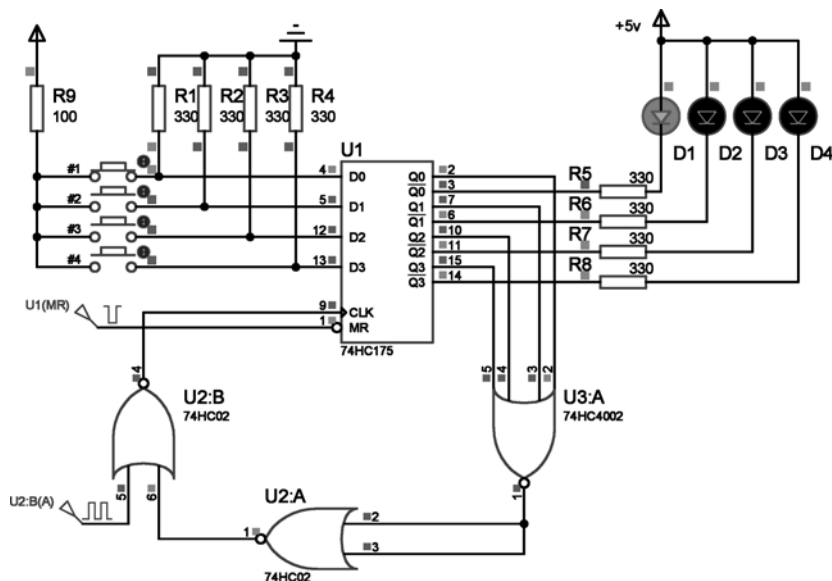
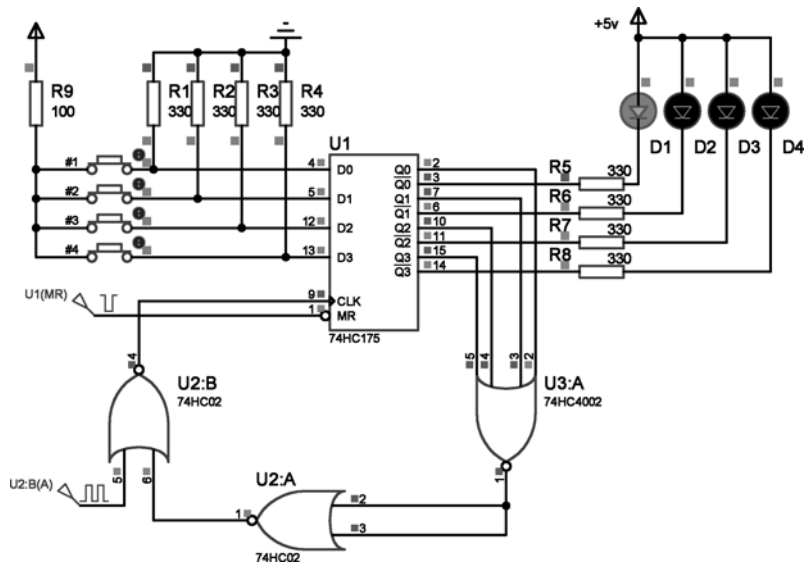


图 5-76 按下#1 键后系统的仿真结果

4. 电路分析

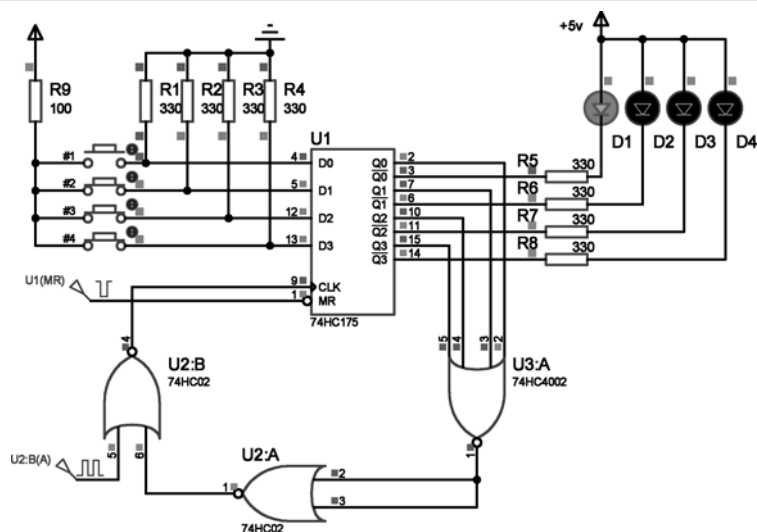
图 5-74 中使用灌电流和或非门来设计。图 5-59 中使用拉电流和与非门设计。拉电流和灌电流是衡量电路输出驱动能力的参数，由于数字电路的输出只有高、低（0、1）两种电平值，高电平输出时，一般是对负载提供电流，其提供电流的数值叫“拉电流”；低电平输出时，一般要吸收负载的电流，其吸收电流的数值叫“灌电流”。

或非门的定义是当输入都为低电平时，输出才为高电平。在按下#1 键时，D0 为高电平，Q0 也为高电平，给予 U3 高电平的输入信号，输出为低电平，经由 U2: A, B 后给时钟信号输入低电平，因此再按其他按键均没有作用，如图 5-77 所示。



(a) #1 键按下状态，系统不响应其他按键的操作

图 5-77 系统不响应其他按键的操作



(b) #1 键释放后，系统仍不响应其他按键的操作

图 5-77 系统不响应其他按键的操作 (续)

第6章 模拟电路设计实例

——音频功率放大器的设计



6.1 设计任务及要求

【设计题目】音频功率放大器设计。

【设计要求】在放大通道的正弦信号输入电压幅度为5~10mV，等效负载电阻 $R_L=8\Omega$ 下，放大通道应满足如下要求：

- ☺ 额定输出功率 $P_{OR} \geq 2W$;
- ☺ 带宽 BW 为20~20 000Hz;
- ☺ 在 P_{OR} 下和 BW 内的非线性失真系数 $\leq 3\%$;
- ☺ 在 P_{OR} 下的效率 $\geq 55\%$;
- ☺ 在前置放大级输入端交流短接到地时， $R_L=8\Omega$ 上的交流声功率 $\leq 10mW$ 。



6.2 音频功率放大器简介

音频功率放大器是声重放设备的重要组成部分，其作用是将传声器件获得的微弱信号放大到足够的强度去推动声重放系统中的扬声器或其他电声器件，使得原声重现。本次设计在仿真软件的基础上，设计一个简单的音频放大器，将小的音频信号放大，并在一定的频率范围内将信号输出。电路的设计包括直流稳压源、音调控制电路、一级放大电路、二级放大电路、工频陷波电路、功率放大电路共六部分，一级、二级放大电路同时充当带通滤波器的角色。其中直流稳压源输出 $\pm 15V$ 的稳定电压，为整个电路提供稳定的电源电压；前级放大电路（包括一级和二级放大电路）的作用是对输入信号进行放大，以满足额定输出功率，其中输入级基本上都用运放作为前置单元；音调控制采用反馈式控制电路，可以调控高低音的放大或衰减程度；工频陷波器主要用来滤除来自稳压电源的工频干扰，提升音频功放的性能；功率放大电路用来输出足够大的功率以驱动扬声器放声。

电路总体框架如图6-1所示。

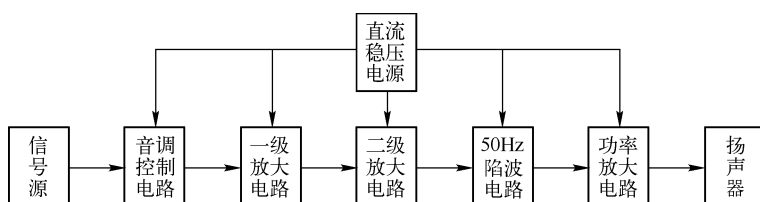


图6-1 音频功放结构



6.3 直流稳压源设计

6.3.1 原理分析与设计

设计好的直流稳压电源电路如图 6-2 所示。

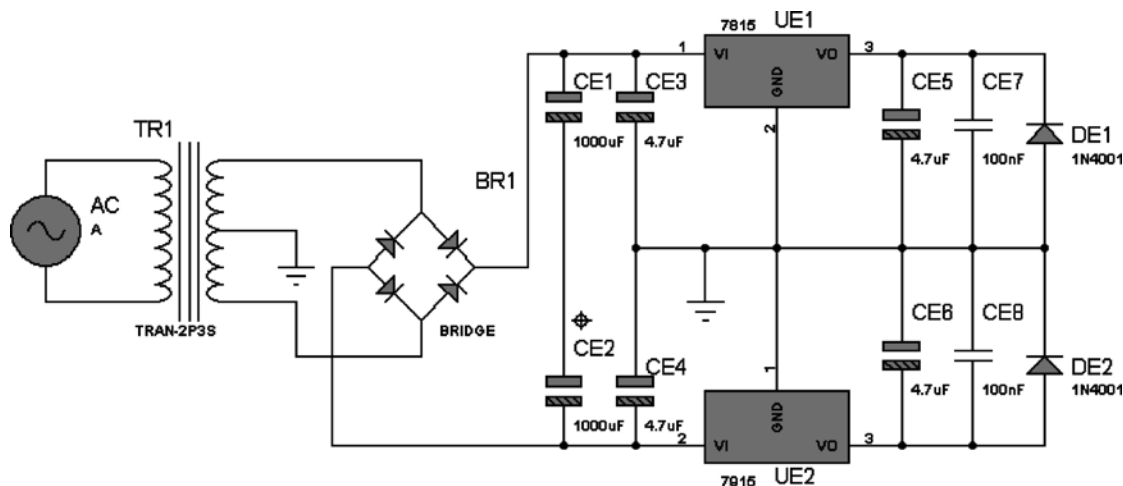


图 6-2 直流稳压电源电路

直流稳压电源的要求是同时输出+15V、-15V 两路电压，最大输出电流为 1A，电压调整率 $\leq 0.2\%$ ，负载调整率 $\leq 1\%$ ，纹波电压（峰-峰值） $\leq 5\text{mV}$ （最低输入电压下，满载），具有过流及短路保护功能。为此，三端稳压器选择 7815、7915（输出电压 $\pm 15\text{V}$ ，最大输出电流 1A，且稳压器内部已有限流电路）。变压器一级线圈与二级线圈之比为 1:5。采用 4 只整流二极管构成全桥整流电路。电路在三端稳压器的输入端接入电解电容 $C_{CE1}=C_{CE2}=1\,000\mu\text{F}$ ，用于电源滤波，其后并入电解电容 $C_{CE3}=C_{CE4}=4.7\mu\text{F}$ 用于进一步滤波。在三端稳压器输出端接入电解电容 $C_{CE5}=C_{CE6}=4.7\mu\text{F}$ 用于减小电压纹波，而并入陶瓷电容 $C_{CE7}=C_{CE8}=100\text{nF}$ 用于改善负载的瞬态响应并抑制高频干扰（陶瓷小电容电感效应很小，可以忽略，而电解电容因为电感效应在高频段比较明显，所以不能抑制高频干扰）。

在输出端同时并入二极管 DE1、DE2（型号为 1N4001），当三端稳压器未接入输入电压时可保护其不至于损坏。

6.3.2 计算机仿真分析

1. 输出端电压

按照图 6-2 所示编辑元件参数，并在正负输出端分别放置电压探针。放置模拟分析图表，设置图表的终止仿真时间为 50ms。添加探针到图表中，仿真结果如图 6-3 所示。

由上图可知，正电压为 15.0V，负电压为 -15.1V，基本符合设计要求，负电压绝对值偏

大是因为三端稳压器 7815 和 7915 各参数不可能绝对对称。

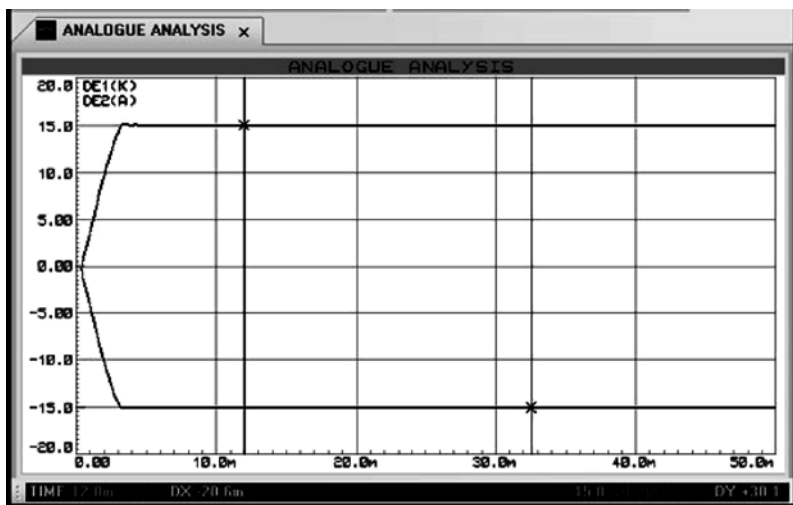



图 6-3 直流电源输出端电压

2. 最大输出电流

通过改变负载电阻, 可以得出最大输出电流。添加电压和电流探针, 注意电流探针的电流方向。单击仿真按钮 , 仿真结果如图 6-4 所示。

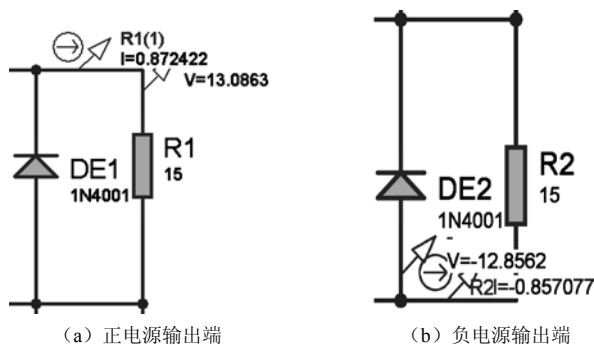


图 6-4 最大输出电流

正负电压端最大输出电流分别如图 6-4 (a), (b) 所示。可见, 没有达到预期的最大输出电流 1A, 正电压端为 0.87A, 负电压端为 0.86A, 不过对于额定功率 2W 而言, 应有的最大电流为 0.7A, 所以这样的最大输出电流可以满足要求。

3. 电压调整率

当输出电流为 500mA 时, 按电网电压的波动范围为 $\pm 10\%$ 计算电压调整率。当电网电压为 $220 \times 1.1 = 242\text{V}$ (有效值) 时, 输出电压如图 6-5 所示。

此时, 正电压端输出电压为 14.914 1V, 负电压端输出电压为 -14.982 2V。

当电网电压为 $220 \times 0.9 = 198\text{V}$ (有效值) 时, 输出电压如图 6-6 所示。

此时, 正电压端输出电压为 14.913 7V, 负电压端输出电压为 -14.980 1V。

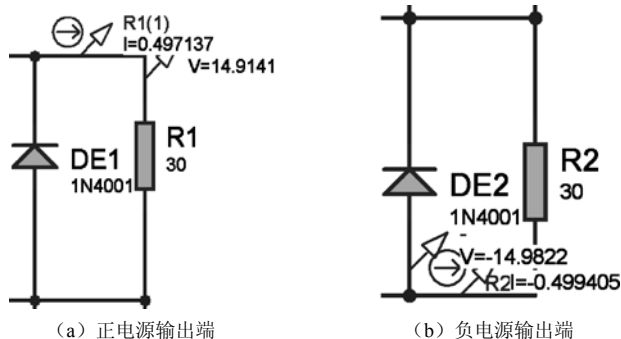


图 6-5 测量电压调整率（电网电压偏高时）

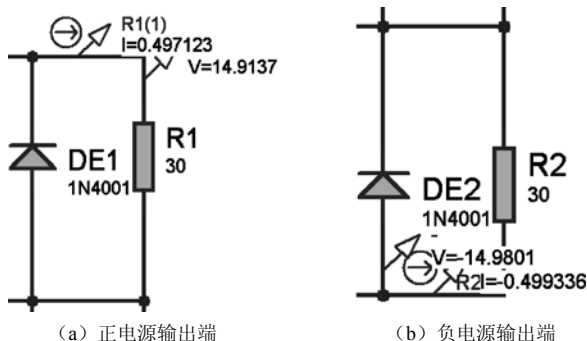


图 6-6 测量电压调整率（电网电压偏低时）

故正电压调整率为

$$S_{U+} = \frac{14.9141 - 14.9137}{15} = 0.002\%$$

负电压调整率为

$$S_{U-} = \frac{14.9822 - 14.9801}{15} = 0.014\%$$

它们均低于所要求的值 0.2%。

4. 电流调整率

在电网电压为 220V，负载电流从 10mA 变为 500mA 时，测量该直流电源的电流调整率。当负载电流为 10mA 时，输出电压如图 6-7 所示。

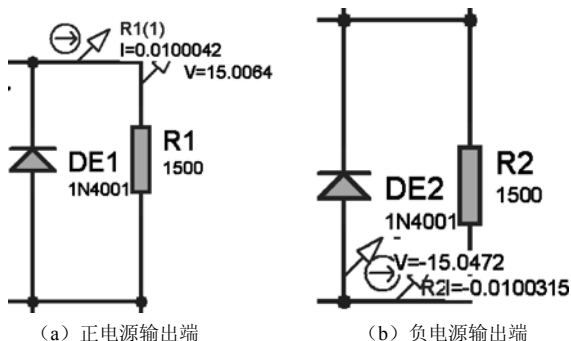


图 6-7 测量电流调整率（负载电流为 10mA）

此时, 电路的正电压端输出电压为 15.006 4V, 负电压端输出电压为-15.047 2V。
当负载电流为 500mA 时, 输出电压如图 6-8 所示。

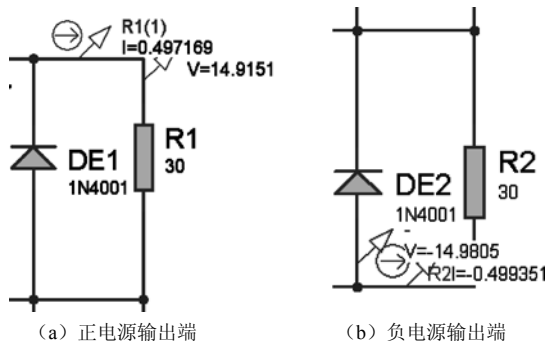


图 6-8 测量电流调整率 (负载电流为 500mA)

此时, 正电压端输出电压为 14.915 1V, 负电压端输出电压为-14.980 5V。
于是可计算出该电源电路的正电压负载调整率为

$$S_{I+} = \frac{15.006\ 4 - 14.915\ 1}{15} = 0.61\%$$

负电压负载调整率为

$$S_{I-} = \frac{15.047\ 2 - 14.980\ 5}{15} = 0.44\%$$

它们均低于所要求的电压调整率 1%。

5. 纹波电压

用模拟分析分析电路的输出电压, 如图 6-9 所示。

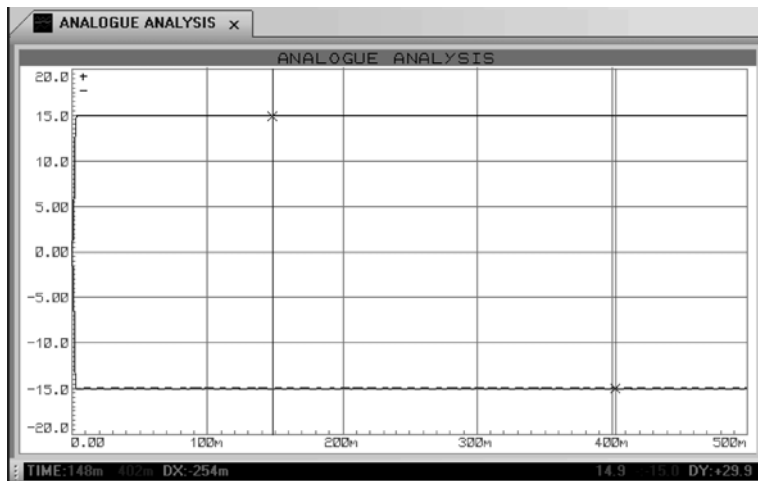


图 6-9 稳压电路的输出电压

结果表明, 在软件设定的精度内, 在 mV 级别上纹波电压几乎为零, 输出电压较为稳定。
再利用傅里叶分析观察正电压端输出电压的谐波分量, 如图 6-10 所示。

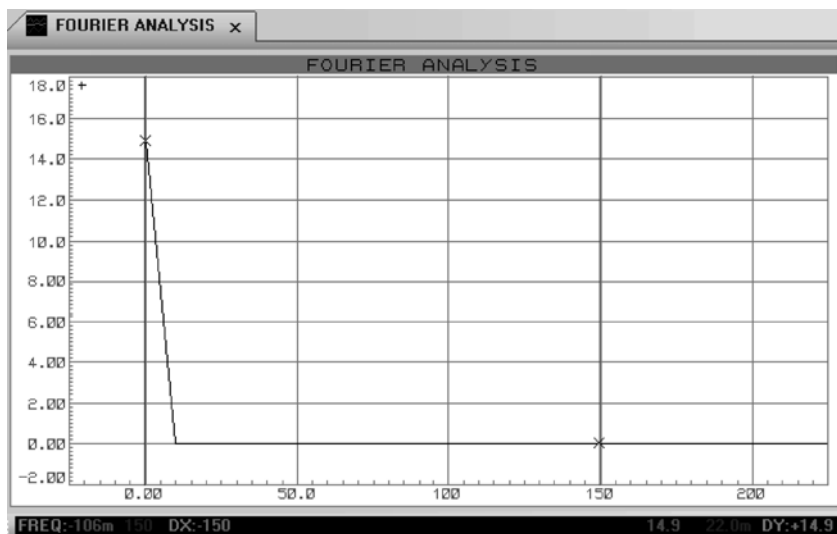


图 6-10 正电压端输出电压的傅里叶分析

由上图可知电源电路在 0Hz 处的分量幅度即直流分量为+15V，而在其他频率点的分量为零。负电压端输出电压的情况类似，如图 6-11 所示。

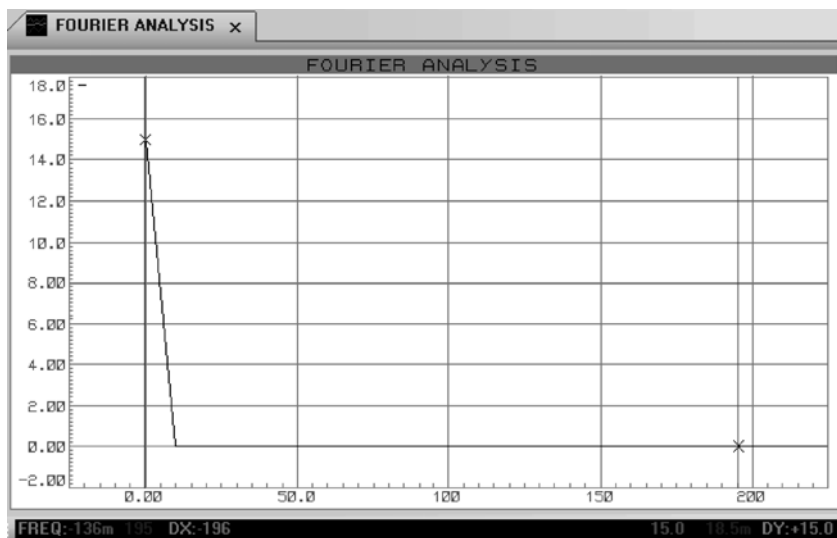


图 6-11 负电压端输出电压的傅里叶分析



6.4 音调控制电路

高低音控制器是高传真放音设备中不可缺少的单元，它能突出或削弱某段声频范围，以满足听者对不同频率的需要。音调控制电路的形式不少，其中以反馈型电路最为常用，它具有较宽的控制范围和较小的失真等优点，只是电路稍复杂些。与流行方案不同的是，该控制电路中 $R_1 \neq R_2$ ，而是 $R_1 = nR_2$ ，此时音调电路的性能好些。

6.4.1 原理分析与设计

设计好的音调控制电路如图 6-12 所示。

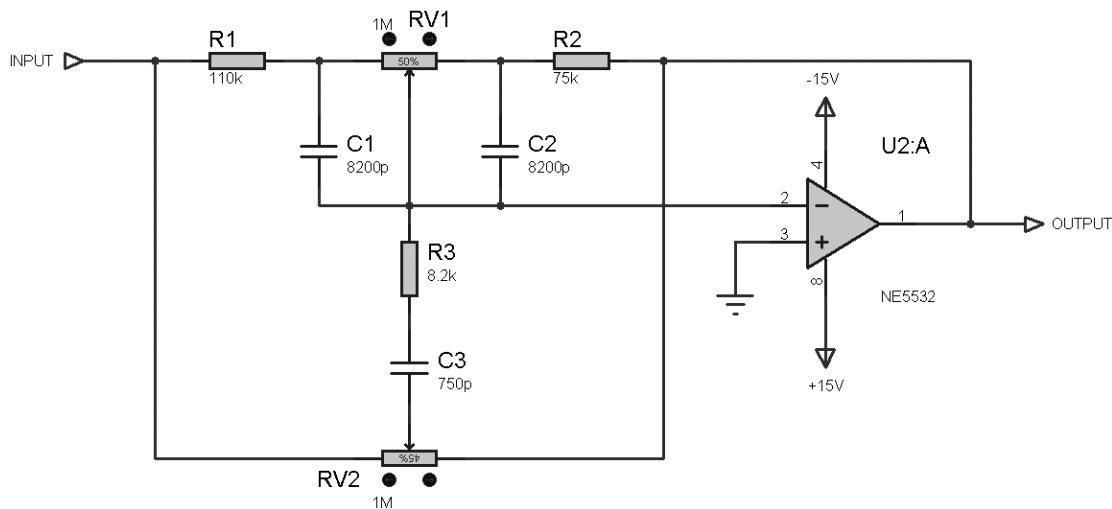


图 6-12 音调控制电路

电路中, RV1、RV2 分别是低音和高音控制电位器。 $C_1=C_2$, $C_3<C_1$, $R_1=nR_2$ 。这些阻容元件构成的网络与运算放大器一起构成了具有高、低频两条反馈通道的滤波器, 分别调节图中的两个电位器, 便可测得增益随频率变化的音调控制曲线。

该音调控制电路高低音最大提升时幅频特性曲线的四个转折频率分别为

$$f_1 = \frac{1}{2\pi RV_1 C_2} \quad (6-1)$$

$$f_2 = \frac{1}{2\pi (RV_1 // R_2) C_2} \quad (6-2)$$

$$f_3 = \frac{1}{2\pi (R_1 + R_3) C_3} \quad (6-3)$$

$$f_4 = \frac{1}{2\pi R_3 C_3} \quad (6-4)$$

其中, f_1 、 f_4 是控制电路最大提升曲线两端的转折频率, f_2 、 f_3 是中间的两个转折频率。

高低音最大衰减时其幅频特性曲线的四个转折频率为

$$f_1 = \frac{1}{2\pi RV_1 C_1} \quad (6-5)$$

$$f_2' = \frac{1}{2\pi (RV_1 // R_1) C_1} \quad (6-6)$$

$$f_3' = \frac{1}{2\pi (R_2 + R_3) C_3} \quad (6-7)$$

$$f_4 = \frac{1}{2\pi R_3 C_3} \quad (6-8)$$

相应地, f_1 、 f_4 是控制电路最大衰减曲线两端的转折频率, f_2' 、 f_3' 是中间的两个转折频率。中音增益为

$$K_M = \frac{R_2}{R_1} \quad (6-9)$$

高低音最大提升时增益为

$$K_R = \frac{RV_1 + R_2}{R_1} \quad (6-10)$$

高低音最大衰减时衰减倍数为

$$K_A = n \cdot K_R = \frac{RV_1 + R_2}{R_2} \quad (6-11)$$

为满足高低音的最大提升倍数和衰减倍数分别相等, 得出

$$R_3 = \frac{R_1 \cdot R_2}{RV_1} \quad (6-12)$$

现令 $K_R = 10$, $RV_1 = RV_2 = 1\text{M}\Omega$, $n = \sqrt{2}$, 则 $K_A \approx 14$ 。令 $f_1 = 20\text{Hz}$, $f_4 = 20\text{kHz}$, 由式 (6-11) 得 $R_2 = 76.09\text{k}\Omega$, 取系列值 $R_2 = 77\text{k}\Omega$ 。

由 $R_1 = \sqrt{2}R_2$ 得出 $R_1 = 108\text{k}\Omega$, 取系列值 $R_1 = 110\text{k}\Omega$ 。

由式 (6-12) 得 $R_3 = 8.3\text{k}\Omega$, 取系列值 $R_3 = 8.2\text{k}\Omega$ 。

由式 (6-1) 得 $C_2 = 8\,000\text{pF}$, 取系列值 $C_2 = 8\,200\text{pF}$, 所以 $C_1 = 8\,200\text{pF}$ 。

由式 (6-4) 得 $C_3 = 970\text{pF}$, 取系列值 $C_3 = 1\,000\text{pF}$ 。

由系列值验算所得结果如下:

$$K_M = \frac{R_2}{R_1} = 0.7(-3\text{dB})$$

$$K_R = \frac{RV_1 + R_2}{R_1} = 9.8(19.8\text{dB})$$

$$K_A = \frac{RV_1 + R_1}{R_2} = 14.8(23.5\text{dB})$$

$$n = \frac{R_1}{R_2} = 1.5$$

$$f_1 = \frac{1}{2\pi RV_1 C_2} = 19.4\text{Hz}$$

$$f_2 = \frac{1}{2\pi(RV_1 // R_2) C_2} = 278\text{Hz}$$

$$f_3 = \frac{1}{2\pi(R_1 + R_3) C_3} = 1.33\text{kHz}$$

$$f_3' = \frac{1}{2\pi(R_2 + R_3) C_3} = 1.91\text{kHz}$$

$$f_4 = \frac{1}{2\pi R_3 C_3} = 19.4\text{kHz}$$

上述结果基本符合要求, 即可认为设计基本完毕。可进行电路的仿真分析。

6.4.2 计算机辅助设计与分析

电路添加正弦信号源和电压探针,分别命名为 INPUT 和 OUTPUT。正弦信号的幅值为 1V,频率为 50Hz。当 RV1、RV2 都滑到最左端,高低音都处于最大提升状态时,电路的幅频特性曲线如图 6-13 所示。

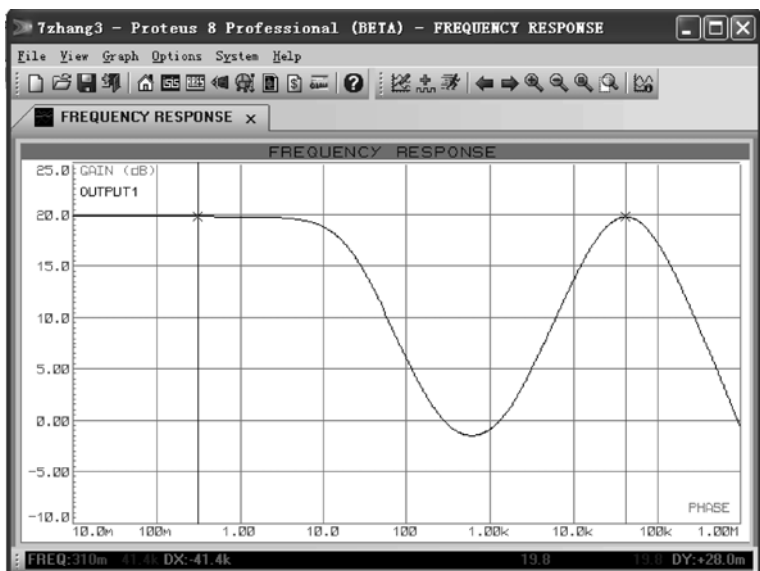


图 6-13 高低音最大提升时的幅频特性曲线

由图 6-13 可知,低音增益为 19.8dB,高音增益为 19.8dB。高音增益偏低及高频段出现大幅衰减是因为运放内晶体管呈现低通特性。

截止频率的测量如图 6-14 所示。

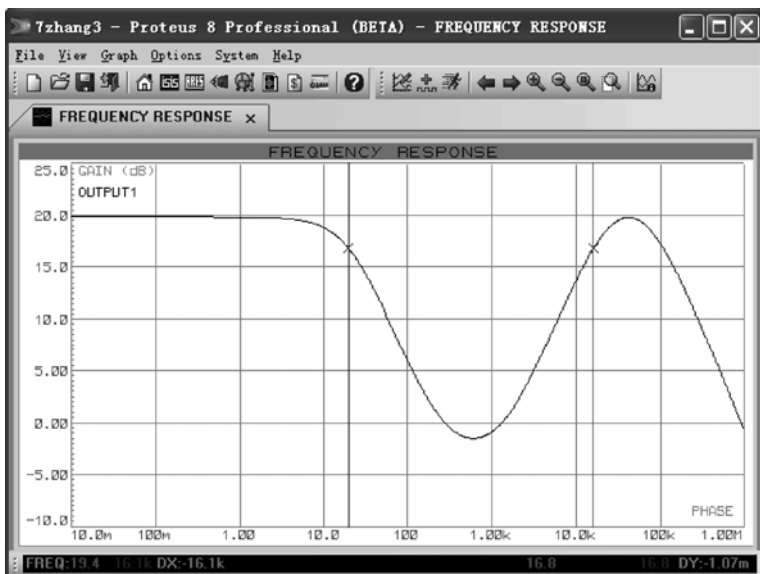


图 6-14 最大提升曲线的截止频率

由图 6-14 可知, $f_1=19.4\text{Hz}$, $f_4=16.1\text{kHz}$ 。上限截止频率偏低还是因为运放的低通特性。由式 (6-4) 知, 通过减小电容 C_3 , 可增大 f_4 , 进而由多级放大电路截止频率的计算公式知, 可增大该音调控制电路的 f_4 。经过调试, 当 $C_3=750\text{pF}$ (为系列值) 时, 幅频特性曲线如图 6-15 所示。

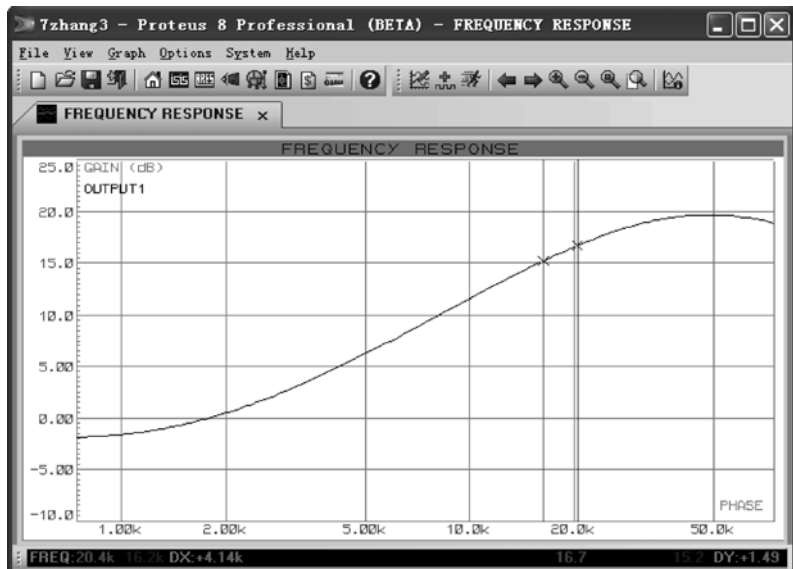


图 6-15 $C_3=750\text{pF}$ 时的幅频特性曲线

此时 $f_4=20.4\text{kHz}$, 基本符合要求。

中音增益的仿真值如图 6-16 所示。

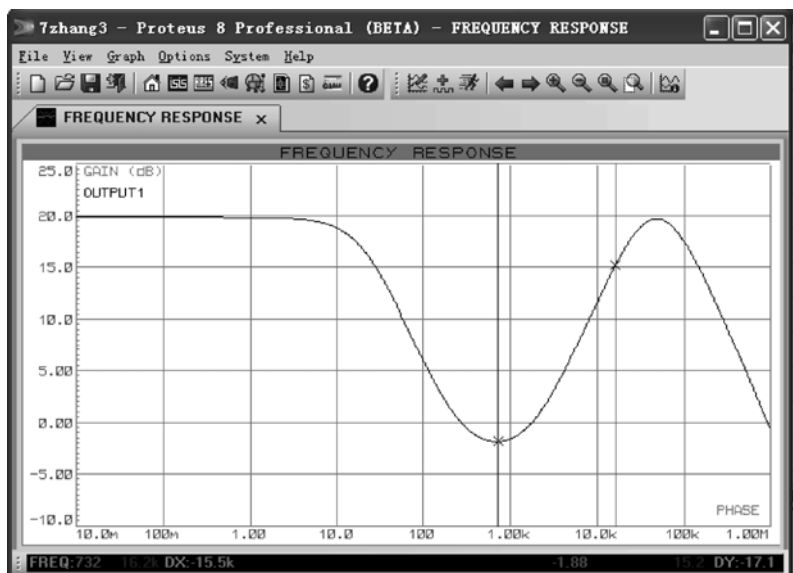


图 6-16 提升曲线的中音增益

中音增益仿真值为 -1.88dB 。该值的计算值 (-3dB) 不很准确, 计算过程采用了近似的分析方法, 与仿真值偏差较大。

当 RV1、RV2 都滑到最右端时高低音处于最大衰减状态，电路的幅频特性曲线如图 6-17 所示。

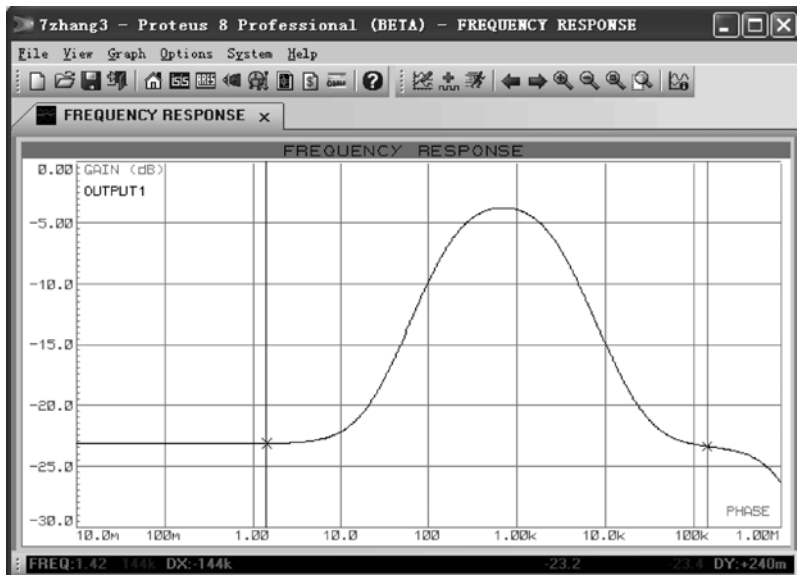


图 6-17 高低音最大衰减时的幅频特性曲线

测得高低音增益为-23.2dB，与计算值相符。在 500kHz 处增益出现剧烈的衰减，同样，这是运放的低通特性所致。

此时的转折频率测量如图 6-18 所示。

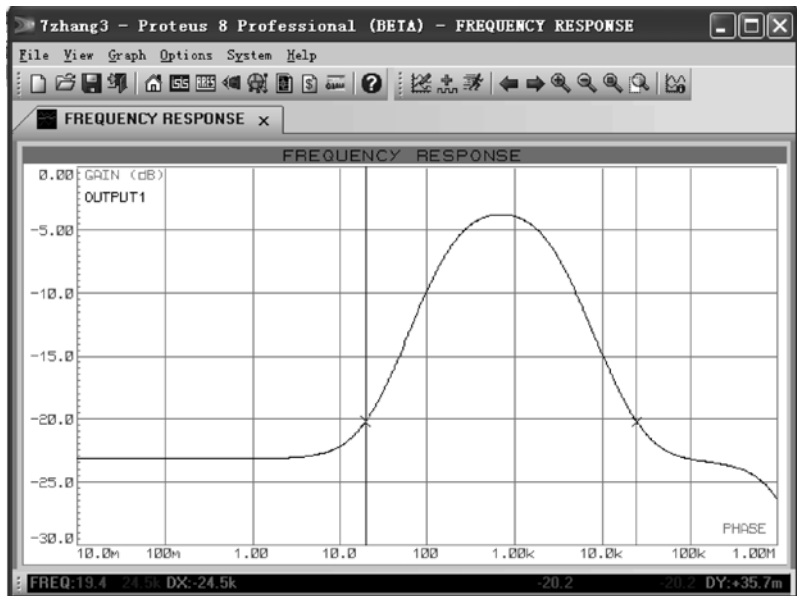


图 6-18 最大衰减曲线的转折频率

由图 6-18 可知， $f_1=19.4\text{Hz}$ ， $f_4=24.5\text{kHz}$ 。在理论推导的过程中得出最大提升曲线的 f_4 和最大衰减曲线的 f_4 相等，但由于各环节的传递函数的作用，此时的 f_4 偏高。

中音增益的测量结果如图 6-19 所示。

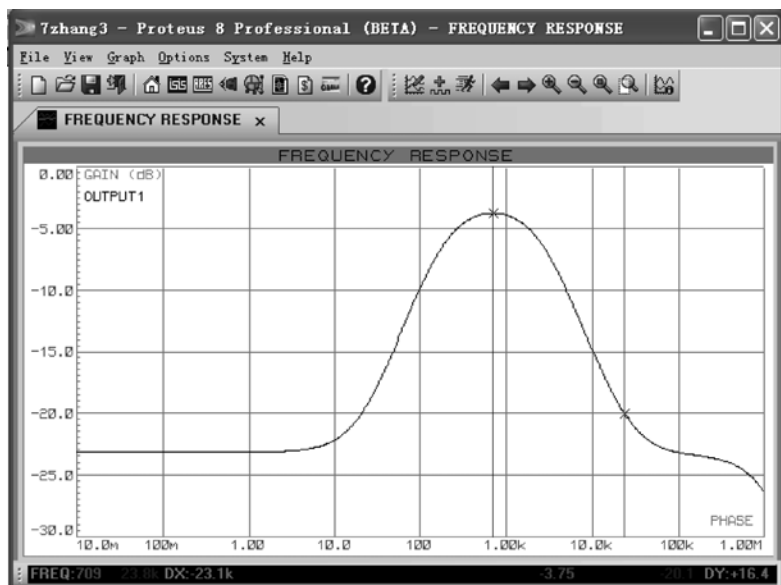


图 6-19 衰减曲线的中音增益

由测量结果可知，中音增益为-3.75dB。

当低音最大提升而高音最大衰减，即 RV1 在最左端，RV2 在最右端时，电路的幅频特性曲线如图 6-20 所示。

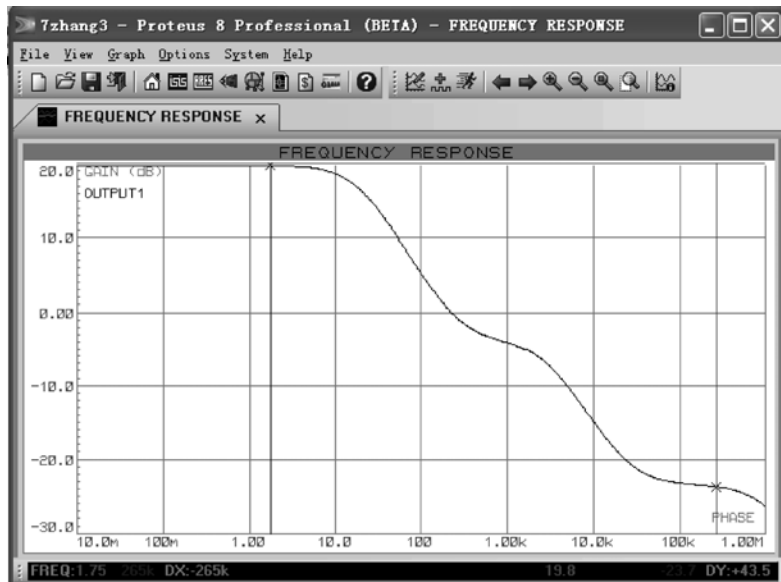


图 6-20 低音最大提升而高音最大衰减时的幅频特性曲线

曲线形状和预期的一致，并且高、中、低音增益， f_1 、 f_4 等几乎不变。

当低音最大衰减而高音最大提升，即 RV1 在最右端，RV2 在最左端时，电路的幅频特性曲线如图 6-21 所示。

同样，曲线的形状和预期的一致，并且高、中音增益， f_1 、 f_4 等几乎不变。

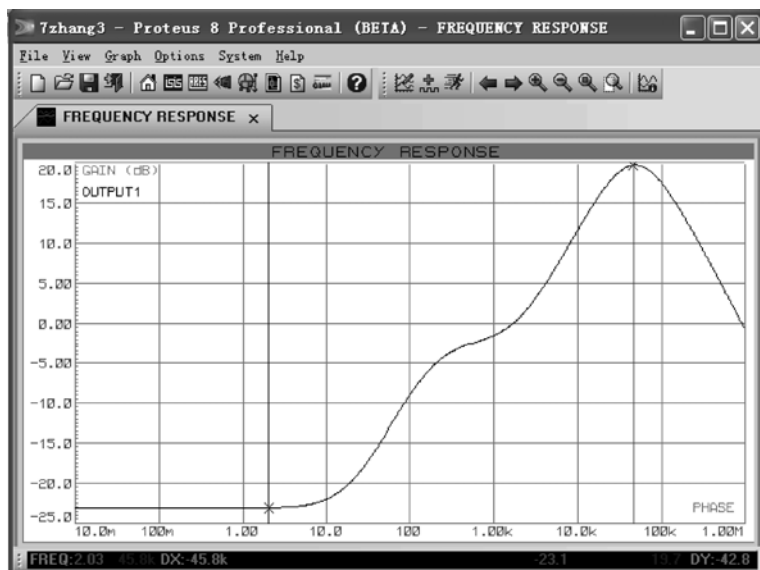


图 6-21 低音最大衰减而高音最大提升时的幅频特性曲线



6.5 工频陷波器

我国市电网供应的电力信号是 50Hz 正弦波, 对于从电网获得工作电源的电子设备存在着 50Hz 工频干扰。为了抑制或减小 50Hz 交流电噪声干扰, 需要在输入电路中加入陷波电路

6.5.1 原理分析与设计

设计好的工频陷波电路如图 6-22 所示。

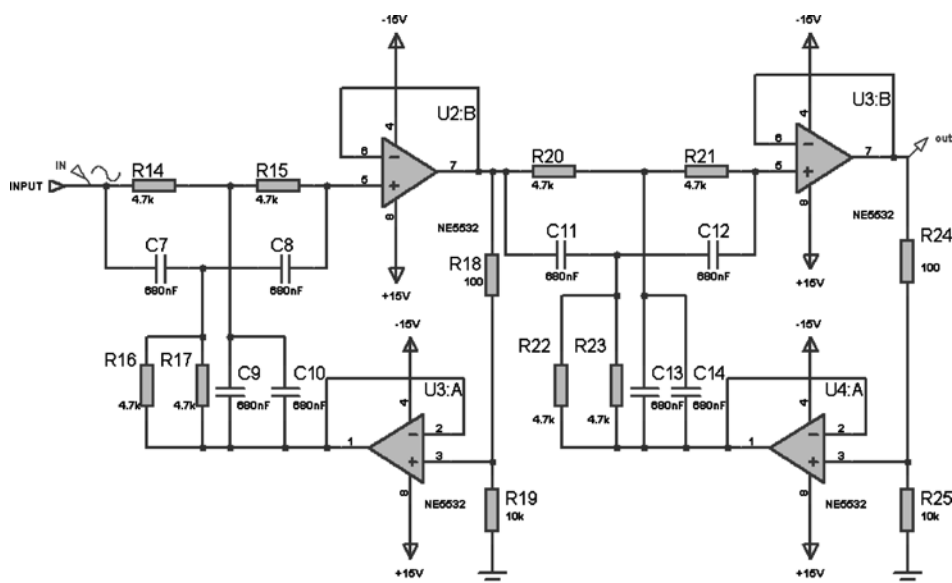


图 6-22 工频陷波电路

电路由两级相同的双 T 双跟随陷波电路级联而成, 目的是让电路在 50Hz 处衰减的程度更大。电路中, $R_{14}=R_{15}=R_{16}=R_{17}$, $C_7=C_8=C_9=C_{10}$, 第二级类推。对每一级陷波器, 其传递函数为

$$\dot{H}(j\omega) = \frac{\omega^2 - \omega_0^2}{\omega^2 - \omega_0^2 - j4\omega_0(1-K)\omega} \quad (6-13)$$

其中

$$K = \frac{R_{19}}{R_{19} + R_{18}} \quad (6-14)$$

中心频率

$$f_0 = \frac{1}{2\pi R_{14} C_7} \quad (6-15)$$

品质因数

$$Q = \frac{1}{4(1-K)} \quad (6-16)$$

阻带宽度

$$BW = \frac{f_0}{Q} \quad (6-17)$$

由上式可知, Q 值越高, 阻带越窄, 有用信号的失真越小。

该电路的通带增益

$$\dot{A}_{up} = 1 \quad (6-18)$$

由于要利用电阻、电容的系列值来组成电路, 使中心频率严格等于 50Hz 将会变得十分困难, 同时电阻、电容的实际值和标称值总存在偏差, 故采用两级陷波器级联的方式, 增大 50Hz 处对信号的衰减程度。通过对 30 多组电阻、电容值对应的中心频率的比较与分析, 最终选定 $R_{14}=4.7k\Omega$, $C_7=680nF$ 。此时, 中心频率

$$f_0 = \frac{1}{2\pi R_{14} C_7} = 49.798\text{Hz}$$

同时, 令 $Q=25$, $R_{19}=10k\Omega$, 由式 (6-14)、式 (6-16) 可得

$$R_{18} = 101.01\Omega$$

取系列值 $R_{18}=100\Omega$ 。此时, $Q=25.25$ 。在以上所选取的参数下, 电路在 50Hz 处的增益计算如下:

$$\omega_0 = 2\pi f_0 = 312.89\text{rad/s} \quad (6-19)$$

$$\omega = 2\pi f = 314.16\text{rad/s} \quad (6-20)$$

$$K = \frac{R_{19}}{R_{19} + R_{18}} = \frac{10\,000}{10\,100} = 0.99 \quad (6-21)$$

将式 (6-19) ~ 式 (6-21) 代入式 (6-13), 得

$$|\dot{H}(j100\pi)| = 0.198(-14.07\text{dB}) \quad (6-22)$$

则电路在该频率处的增益

$$|\dot{A}_u|_{f=50\text{Hz}} = |\dot{H}(j100\pi)|^2 = 0.039(-28.18\text{dB}) \quad (6-23)$$

6.5.2 计算机仿真分析

放置频率分析图表, 添加输出电压探针 OUT, 开始仿真。电路的幅频特性曲线如图 6-23 所示。

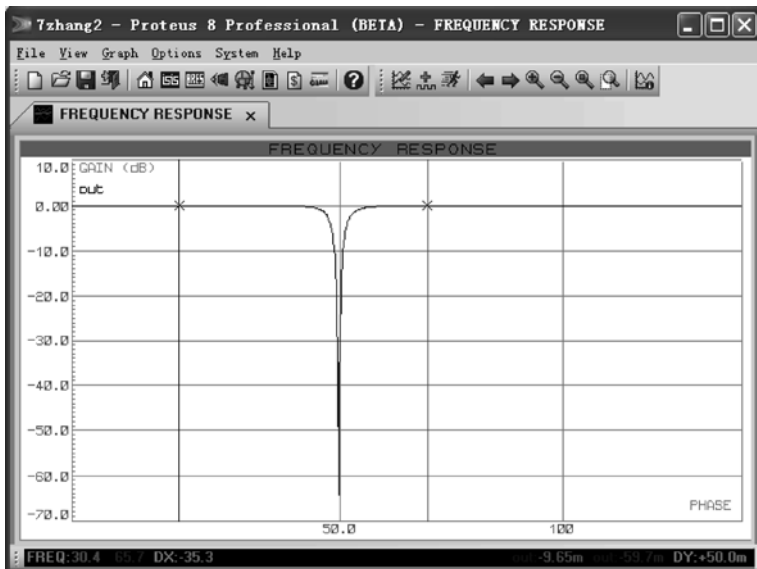


图 6-23 工频陷波器的幅频特性曲线

由指针的读数可以看出, 电路的通带增益几乎为零。图表放大后, 观察电路各参数, 如图 6-24 所示。

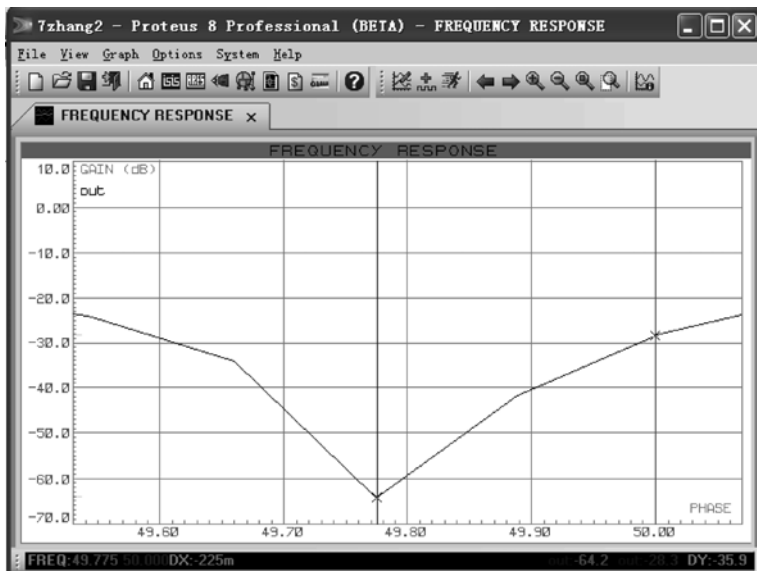


图 6-24 放大后的工频陷波器的幅频特性图表

由测量的结果可知, 陷波器的中心频率为 49.775Hz, 50Hz 处的增益为-28.3dB, 与计算结果相一致。

工频陷波器的阻带测量如图 6-25 所示。

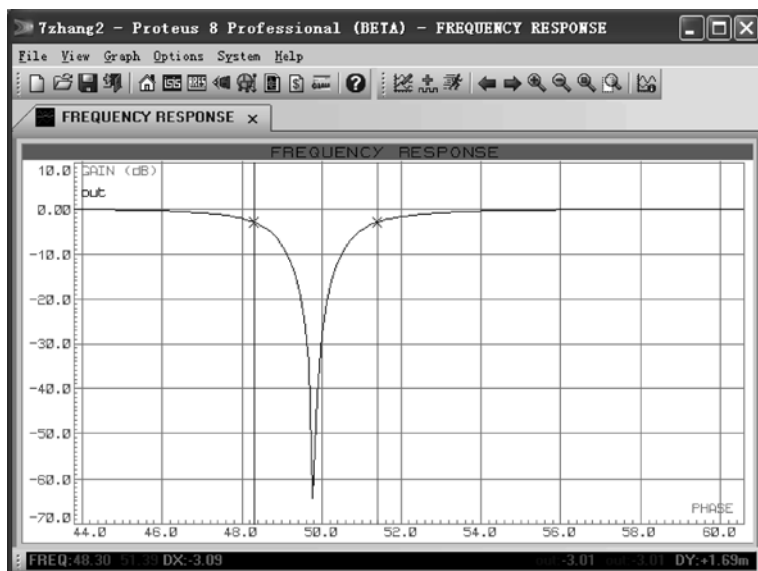


图 6-25 工频陷波器的阻带测量

由测量的结果可知，陷波器的阻带宽为 3.09Hz，截止频率分别为

$$f_{p1} = 48.30\text{Hz}$$

$$f_{p2} = 51.39\text{Hz}$$

这样，各项指标均与理论值相一致，该陷波器的设计较为成功。

分别选择输入正弦波信号为 1V、5Hz，1V、50Hz 和 1V、1kHz 进行分析，来计算各频率处的增益。以上各信号电压均为幅值。

当输入信号为 1V、5Hz 时，电路的输入/输出电压波形如图 6-26 所示。

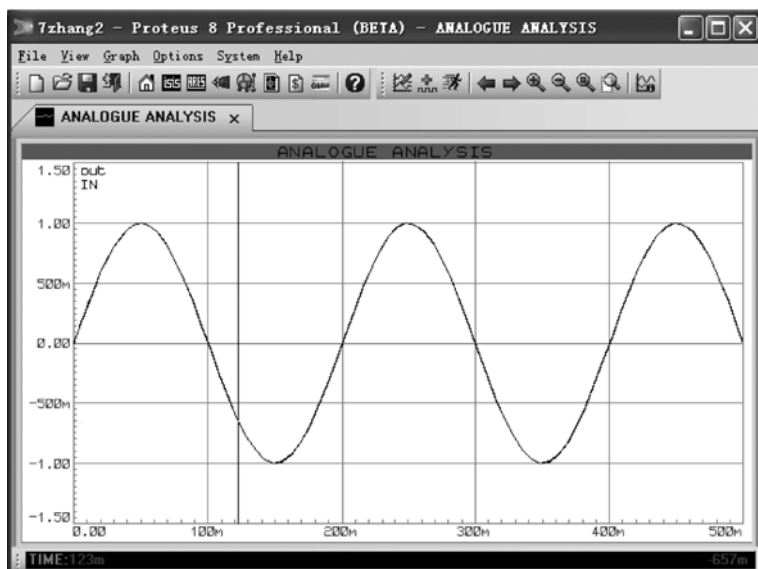


图 6-26 输入信号为 1V、5Hz 时陷波电路的输入/输出电压波形

图中有两条波形曲线, 分别是输入信号和输出信号, 它们几乎重合, 幅值都为 1V, 说明此时相移为零, 放大倍数为 1, 与理论分析及频率分析的结果一致。频率分析的相频特性如图 6-27 所示, 图中示出了 5Hz 处的相移为 -0.463° 。

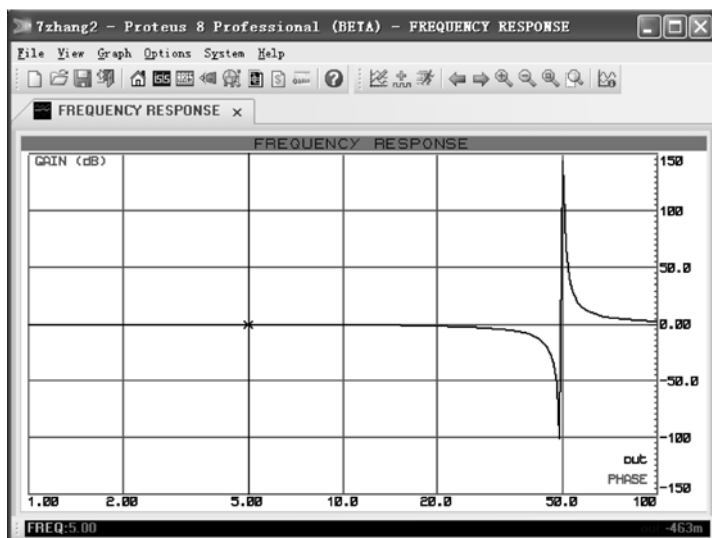


图 6-27 陷波电路在 5Hz 处的相移

当输入信号为 1V、50Hz 时, 电路的输入/输出波形如图 6-28 所示。

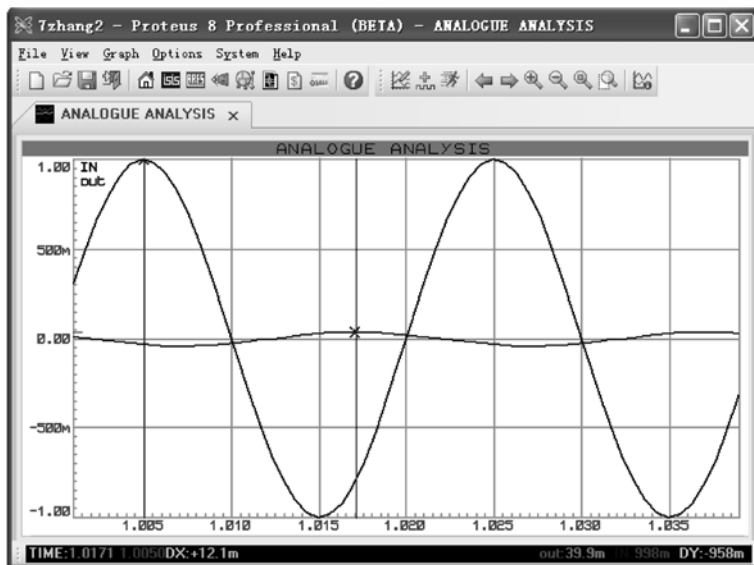


图 6-28 输入信号为 1V、50Hz 时陷波电路的输入/输出波形

输出幅值变为 39.94mV, 放大倍数为

$$\left| \dot{H} \right|_{f=50\text{Hz}} = \frac{0.0399}{1} = 0.0399 (-27.98\text{dB})$$

与理论计算及频率分析的结果几乎相等。

在频率分析中, 50Hz 处相移的测量如图 6-29 所示。

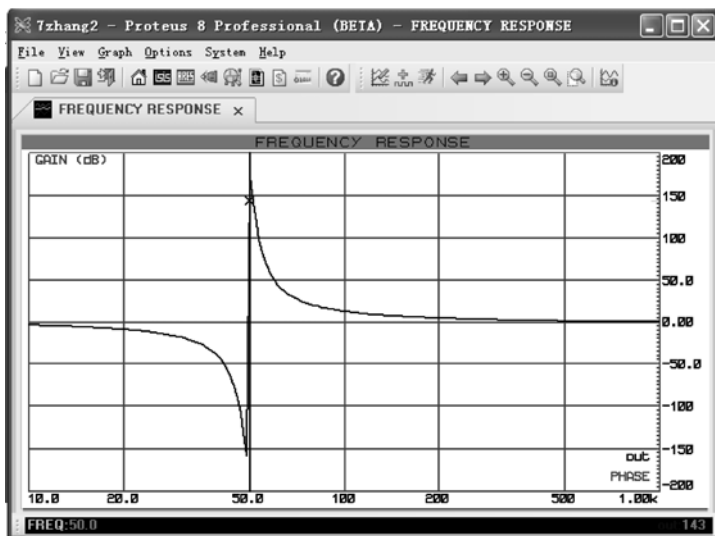


图 6-29 陷波电路在 50Hz 处的相移

由图 6-29 所示的测量结果知 50Hz 处相移为 143° 。而由图 6-30 所示的测量结果可以计算得到电路在 50Hz 处的相移。

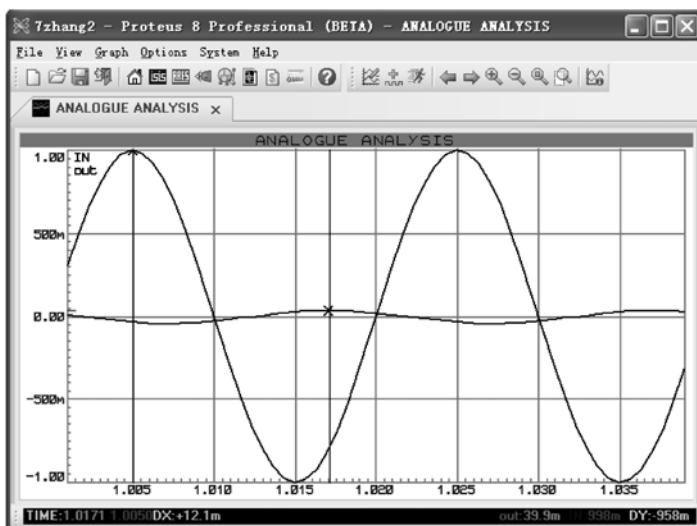


图 6-30 利用模拟分析计算相移

由图 6-30 可得时间差 $\Delta t = 12.1\text{ms}$, 算出相移为

$$\varphi = 12.1\text{ms} \times 100\pi \times \frac{180}{\pi} = 217.8^\circ$$

也即输出超前输入的相角为

$$\varphi' = 360^\circ - 217.8^\circ = 142.2^\circ$$

与频率分析得到的结果相近。

当输入信号为 1V、1kHz 时, 电路的输入/输出波形如图 6-31 所示。

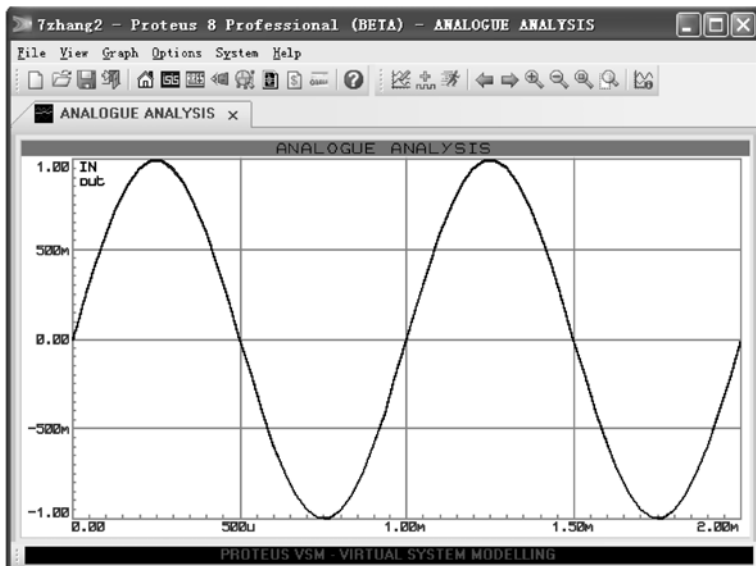


图 6-31 输入信号为 1V、1kHz 时陷波电路的输入/输出波形

同低频 5Hz 时一样, 输入/输出波形几乎重合, 即放大倍数为 1, 且相移为零, 与理论分析及频率分析的结果 (0.112°) 一致。频率分析的结果如图 6-32 所示。

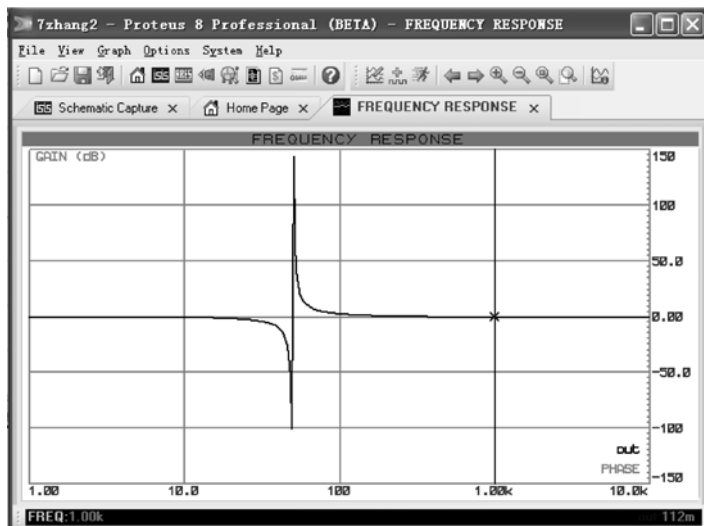


图 6-32 陷波电路在 1kHz 处的相移



6.6 前级放大电路

由于信号源输出电压幅度往往很小, 不足以驱动功率放大器输出额定功率, 因此常在功率放大电路之前插入前置放大器将信号源输出的信号加以放大, 同时对信号进行适当的音色处理。在本次设计中, 前级放大电路 (包括一级放大电路和二级放大电路) 对音色所做的

处理就是滤波的作用，即前级放大电路同时充当带通滤波器的角色，使整体音频功率放大电路的带宽为 20Hz~20kHz。

6.6.1 原理分析与设计

设计好的前级放大电路如图 6-33 所示。

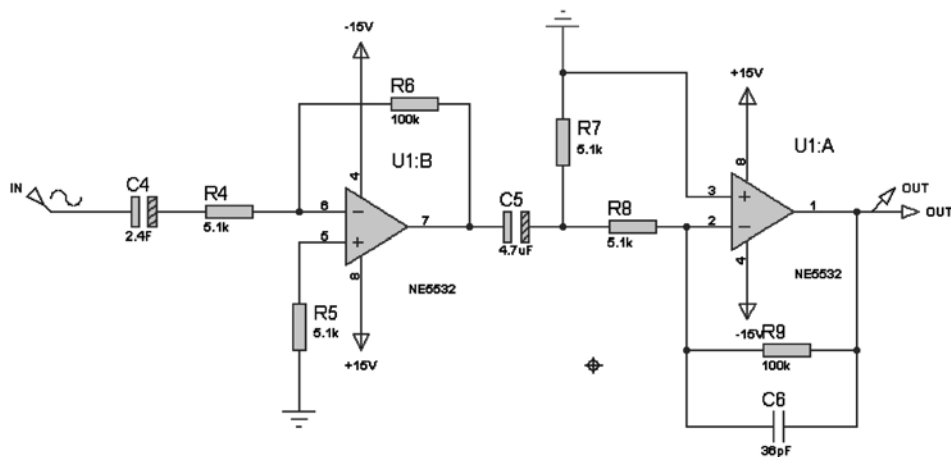


图 6-33 前级放大电路

在电路的通带内，前级放大电路为两级反相比例运算放大电路的级联，故其通带放大倍数为

$$\dot{A}_{up} = \dot{A}_{up1} \cdot \dot{A}_{up2} = \frac{R_6}{R_4} \cdot \frac{R_9}{R_8}$$

在输入信号为 5~10mV 时，为保证额定功率 $P_{OR} \geq 2W$ ，功放输出端电压的峰值应为 5.65V，当功率放大电路的放大倍数最大时（等于 3），前级放大电路的输出电压峰值应为 5.65/3=1.88V，按此时的输入电压为 5mV，那么前级放大电路的放大倍数应至少为 1.88/0.005=376。电路中取 $R_6 = R_9 = 100k\Omega$ ， $R_4 = R_8 = 5.1k\Omega$ ，则放大倍数

$$\dot{A}_{up} = \left(\frac{100}{5.1} \right)^2 = 384.47$$

故放大倍数满足要求。

6.6.2 计算机仿真分析

放置模拟分析图表，设置仿真终止时间为 1.5ms。

当输入信号为 5mV（幅值）、1kHz 的正弦波时，电路的输入/输出波形如图 6-34 所示。

由输入/输出波形可知，电路实现了同相放大（因为是两级反相放大器级联）。由图 6-34 测得输出电压幅度为 1.95V。则电路在 1kHz 处的放大倍数

$$\dot{A}_u = \frac{1.92}{0.005} = 384$$

与理论分析的结果相近。

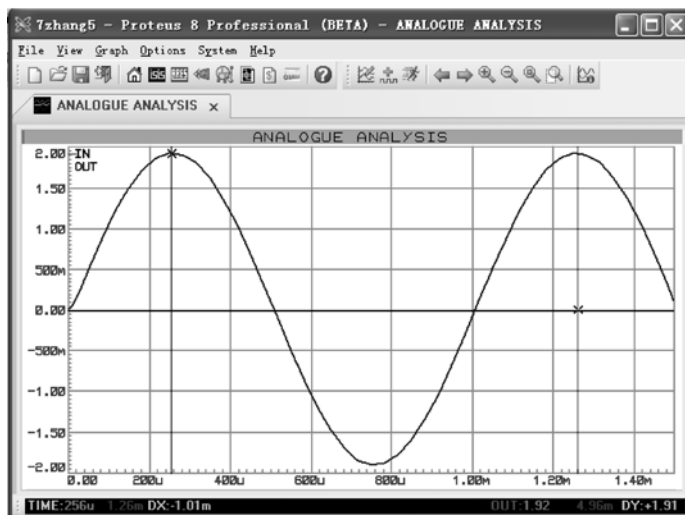


图 6-34 输入信号为 5mV、1kHz 的正弦波时电路的输入/输出波形



6.7 功率放大电路

欲使扬声器发声，必须要用足够的功率来驱动。功率放大器不是仅仅的电压放大或是电流放大，而是追求功率的放大。本次设计采用的是 OCL 功率放大电路。在电路中引入负反馈，可以减小非线性失真、展宽频带。

6.7.1 原理分析与设计

设计好的功率放大电路如图 6-35 所示。

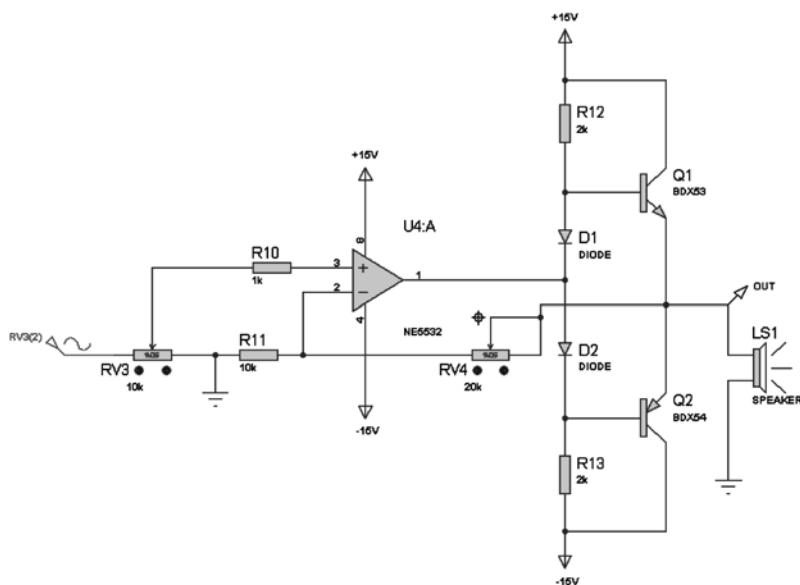


图 6-35 功率放大电路

静态时，正负电源的作用使晶体管 Q_1 、 Q_2 处于微导通状态。在输入信号的正半周主要是 Q_1 管发射极驱动负载，而负半周主要是 Q_2 管发射极驱动负载，而且两管的导通时间都比半个周期长，所以即使输入电压很小，总能保证至少有一只晶体管导通，因而消除了交越失真。

电路最大不失真输出电压的有效值

$$U_{om} = \frac{V_{CC} - U_{CES1}}{\sqrt{2}} \quad (6-24)$$

设饱和管压降

$$U_{CES1} = -U_{CES2} = U_{CES} \quad (6-25)$$

最大输出功率

$$P_{om} = \frac{U_{om}^2}{R_L} = \frac{(V_{CC} - U_{CES})^2}{2R_L} \quad (6-26)$$

在忽略基极回路电流的情况下，电源 V_{CC} 提供的电流

$$i_C = \frac{V_{CC} - U_{CES}}{R_L} \sin \omega t$$

电源在负载获得最大交流功率时所消耗的平均功率等于其平均电流与电源电压之积，其表达式为

$$P_V = \frac{1}{\pi} \int_0^\pi \frac{V_{CC} - U_{CES}}{R_L} \sin \omega t \cdot V_{CC} d\omega t$$

可得

$$P_V = \frac{2}{\pi} \cdot \frac{V_{CC}(V_{CC} - U_{CES})}{R_L} \quad (6-27)$$

因此，转换效率

$$\eta = \frac{P_{om}}{P_V} = \frac{\pi}{4} \cdot \frac{V_{CC} - U_{CES}}{V_{CC}} \quad (6-28)$$

在理想情况下，即饱和管压降可忽略不计的情况下

$$P_{om} = \frac{U_{om}^2}{R_L} = \frac{V_{CC}^2}{2R_L} \quad (6-29)$$

$$P_V = \frac{2}{\pi} \cdot \frac{V_{CC}^2}{R_L} \quad (6-30)$$

$$\eta = \frac{\pi}{4} \approx 78.5\% \quad (6-31)$$

显然，在实际情况中一般都不能忽略饱和管压降，即不能用式（6-29）和式（6-31）计算电路的最大输出功率和效率。在仿真中测得所用的晶体管饱和管压降大约为 $U_{CES} = 3.1V$ 。故由式（6-26）算得最大输出功率

$$P_{om} = 8.85W$$

满足额定功率 $P_{OR} \geq 2W$ 的要求。

此时的效率

$$\eta = 62.3\%$$

故满足在 P_{OR} 下的效率 $\geq 55\%$ 的要求。

6.7.2 计算机仿真分析

通过改变输入电压的幅度, 来观察电路的输出, 进而确定功放电路的最大不失真输出电压。调节两个电位器, 使 RV3 为 92%, RV4 为 5%, 使电路的电压放大倍数为 1。

当输入信号为 8V (峰值)、1kHz 的正弦波时, 电路的输出电压波形如图 6-36 所示。

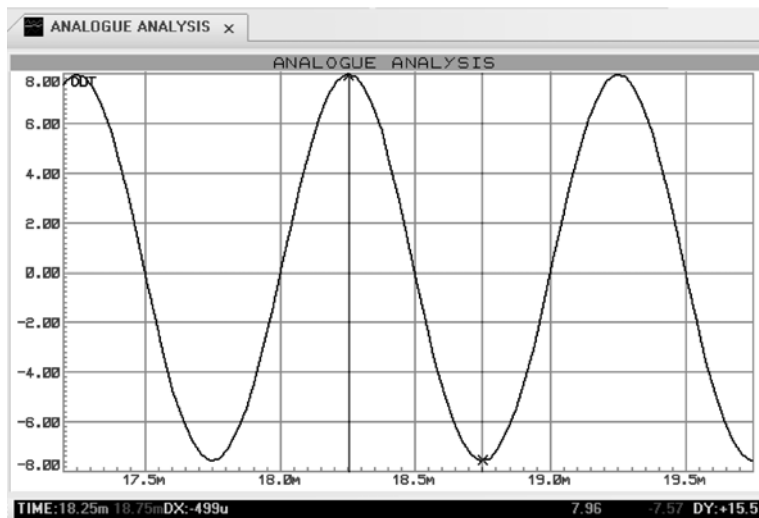


图 6-36 输入信号为 8V、1kHz 时功放电路的输出电压波形

由图 6-36 的测量结果可知, 输出电压的峰值为 6.93V, 并且在负半周出现轻微的饱和失真。令输入信号为 8.8V、1kHz 的正弦信号, 此时电路的输出电压波形如图 6-37 所示。

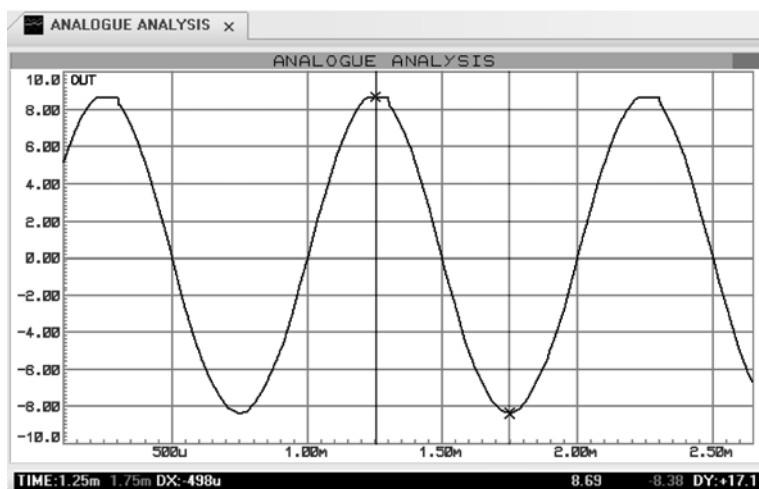


图 6-37 输入信号为 8.8V、1kHz 的正弦信号时电路的输出电压波形

由图 6-37 的测量结果可知, 正电压可能出现的最大峰值为 8.68V, 而负电压可能出现的最大峰值为 8.49V。这是由于互补晶体管的特性不可能绝对对称。

当输入信号为 8.7V、1kHz 时，电路的输出波形如图 6-38 所示。

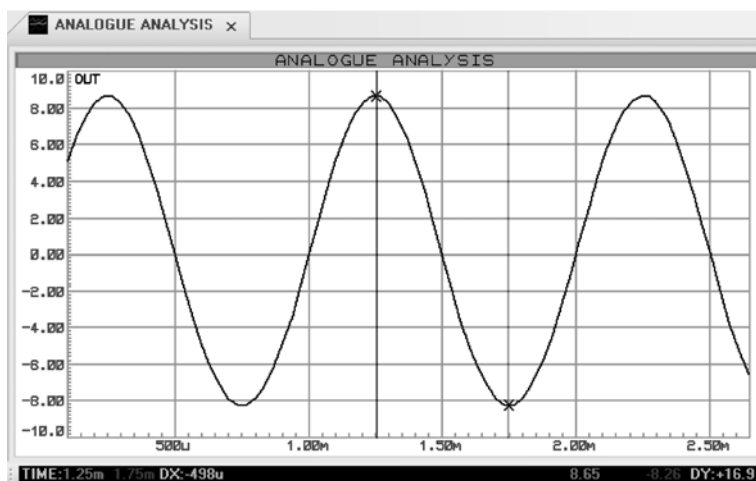


图 6-38 输入信号为 8.7V、1kHz 时功放电路的输出波形

可以认为电路的最大不失真输出电压为 8.7V。这样由式 (6-26) 及式 (6-28) 知最大输出功率和效率与计算值相等。

当输入信号为 1V、1kHz 时，对输出端电压进行傅里叶分析，如图 6-39 所示。

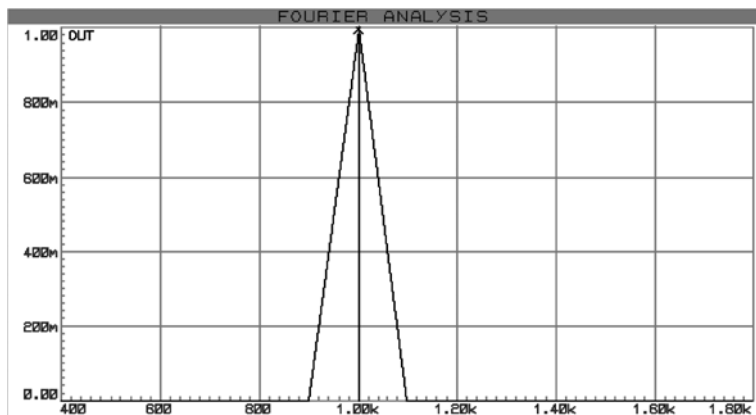


图 6-39 输入信号为 1V、1kHz 时输出电压的傅里叶分析

由图像可得基波的频率为 1kHz，基波的增益大小为 0.95V，二次谐波的增益大小为 0.000 021V，三次谐波的增益大小为 0.000 0012V，又因为整个系统的失真度为

$$\text{THD} = \sqrt{\frac{V_{\text{om}2}^2 + V_{\text{om}3}^2}{V_{\text{om}1}^2}}, \text{ 其中的 } V_{\text{om}1} \text{ 为基波的增益分量, } V_{\text{om}2} \text{ 为二次谐波的增益分量, } V_{\text{om}3}$$

为为三次谐波分量，因此整个系统的总谐波失真 $\text{THD}=0.002\ 214\ 132\%$ 。

此时，能量主要集中在基频 1kHz 处，并且 $\text{THD}=0.002\ 214\ 132\%$ 。这时的总谐波失真很小，可以忽略。

当输入信号为 4V、1kHz 时，对输出端电压进行傅里叶分析，如图 6-40 所示。此时，

THD 变为 0.167 308%，与 1V 时相比，有所增大。说明输出电压越接近最大输出电压，THD 越大。

当输入信号为 8V、1kHz 时，对输出端电压进行傅里叶分析，如图 6-41 所示。果然，THD 变为 1.506 021%，与上一种情况相比，进一步变大。

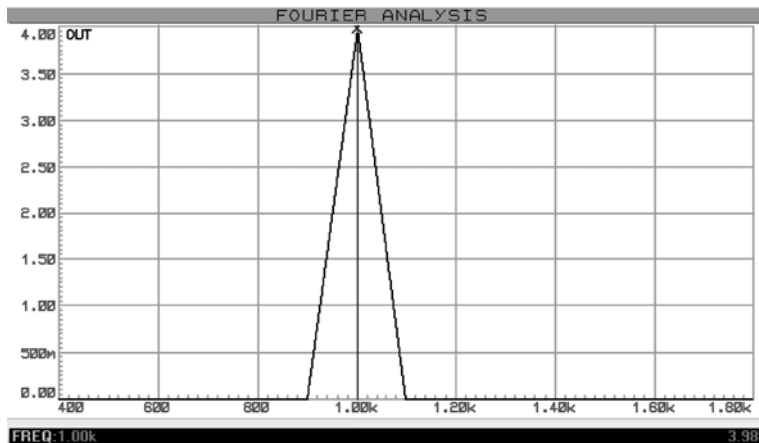


图 6-40 输入信号为 4V、1kHz 时输出电压的傅里叶分析

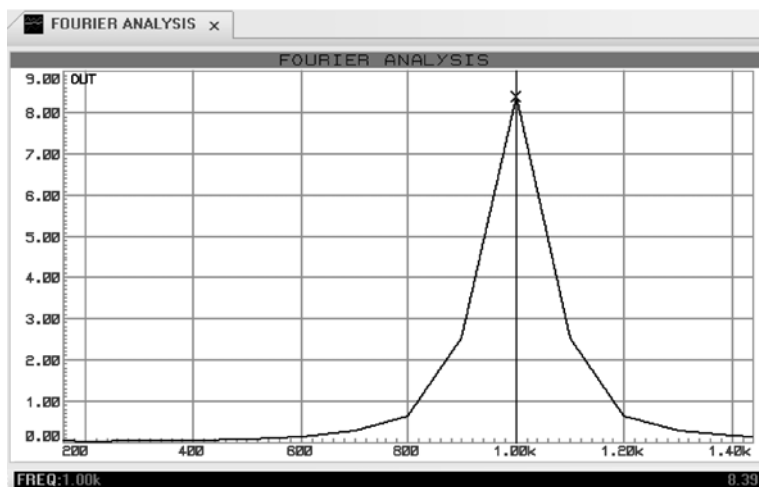


图 6-41 输入信号为 8V、1kHz 时输出电压的傅里叶分析



6.8 电路整体的协调及仿真

6.8.1 电路各组成部分的协调连接

当电路各部分设计完毕后，需对各部分进行适当的连接，并考虑器件间相互的影响。

各部分的连接顺序为：信号源→音调控制电路→一级放大电路→二级放大电路→工频陷波电路→功率放大电路。

其中工频陷波器放在二级放大电路和功率放大电路之间, 是为了最大限度地对来自电源的工频干扰进行抑制。

6.8.2 带通滤波器的加入

在前级放大电路中, 电容与电阻的适当组合构成了带通滤波器。前级放大电路如图 6-42 所示。

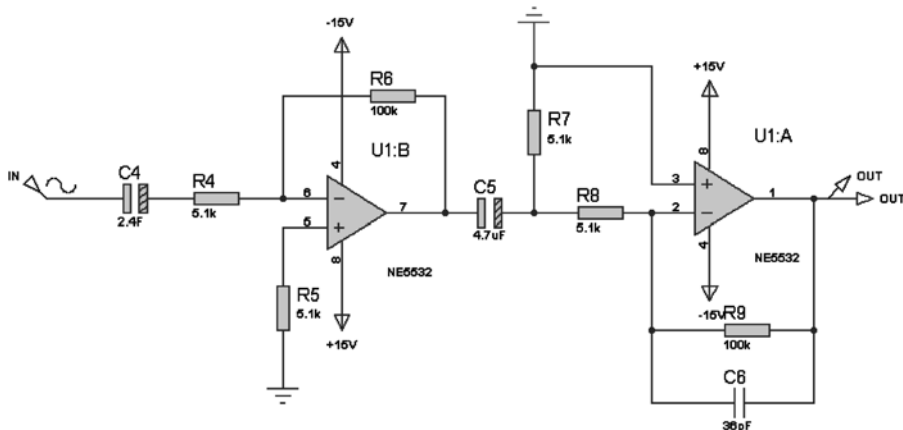


图 6-42 前级放大电路

电路中, R_4 、 C_4 和 C_5 、 R_7 、 R_8 都为高通网络, 决定电路的下限截止频率 f_L 。

$$f_L = 1.1\sqrt{f_{L1}^2 + f_{L2}^2} \quad (6-32)$$

其中,

$$f_{L1} = \frac{1}{2\pi R_4 C_4} \quad (6-33)$$

$$f_{L2} = \frac{1}{2\pi (R_7 // R_8) C_5} \quad (6-34)$$

令 $f_L = 20\text{Hz}$, $f_{L1} = f_{L2}$, 则式 (6-32) 变为

$$f_L = 1.1\sqrt{2}f_{L1} \quad (6-35)$$

得到

$$f_{L1} = 12.86\text{Hz}$$

由式 (6-33) 可得

$$C_4 = \frac{1}{2\pi R_4 f_{L1}} = 2.43\mu\text{F}$$

由式 (6-34) 可得

$$C_5 = \frac{1}{2\pi (R_7 // R_8) f_{L2}} = 4.85\mu\text{F}$$

分别取系列值 $C_4 = 2.4\mu\text{F}$, $C_5 = 4.7\mu\text{F}$ 。将此二值代入式 (6-32), 得

$$f_L = 1.1\sqrt{f_{L1}^2 + f_{L2}^2} = 20.44\text{Hz}$$

同时, R_9 、 C_6 为低通网络, 会影响电路的上限截止频率 f_H 。

$$\frac{1}{f_H} = 1.1 \sqrt{\frac{1}{f_{H1}^2} + \frac{1}{f_{H2}^2}} \quad (6-36)$$

其中, f_{H1} 为 R_9 、 C_6 所决定的上限截止频率, 且

$$f_{H1} = \frac{1}{2\pi R_9 C_6} \quad (6-37)$$

f_{H2} 为运放、晶体管所决定的上限截止频率。在本次设计中, 它的确定将变得非常困难。

在 $R_9 = 100\text{k}\Omega$ 的前提下, 令 $f_{H1} = 20\text{kHz}$, 则可得到

$$C_6 = \frac{1}{2\pi R_9 f_{H1}} = 79.58\text{pF}$$

但如果 C_6 按上述取值, 由式 (6-36) 知电路的上限截止频率 f_H 必然小于 20kHz 。故利用计算机辅助设计的手段确定 C_6 的取值。

6.8.3 计算机辅助设计与分析

在进行仿真分析时, 由于音调控制电路的变化范围大, 传递函数复杂, 故仅研究剩余部分电路的频率响应, 这里只使用了一级放大电路。

当 $C_6 = 79.58\text{pF}$ 且功率放大电路的电压放大倍数为 1 时, 该剩余部分电路的幅频特性曲线如图 6-43 所示。

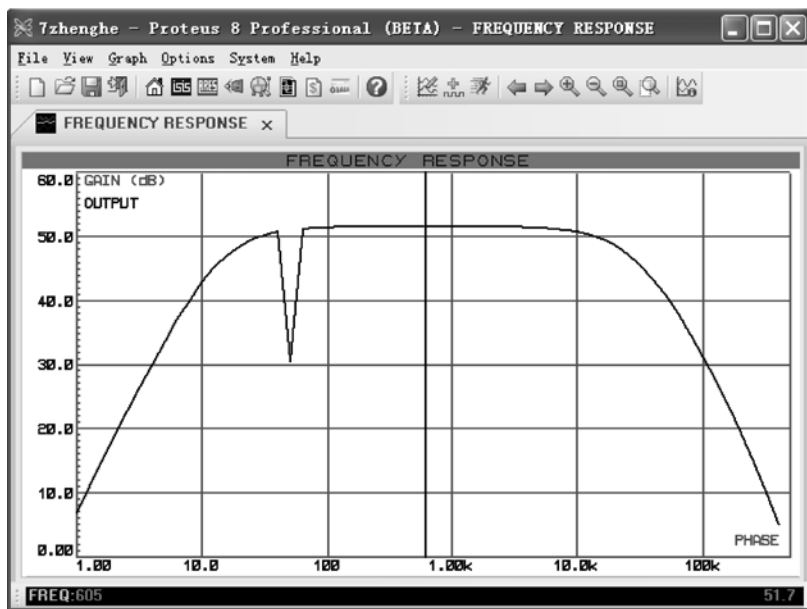


图 6-43 不含音调控制电路时音频功放的幅频特性曲线

由图 6-43 的测量值可得, 电路的通带增益为 51.7dB 。在 6.6.1 节算得通带放大倍数为

$$\dot{A}_{up} = \left(\frac{100}{5.1} \right)^2 = 384.47 (51.697\text{dB})$$

故该仿真值与理论值符合得很好。

电路的通频带测量如图 6-44 所示。

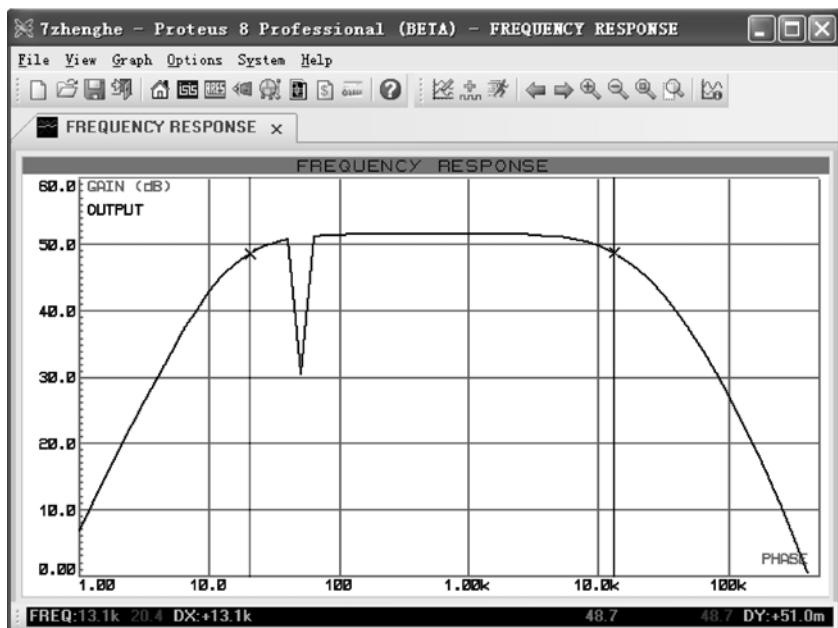


图 6-44 不含音调控制电路时音频功放的通频带

由图 6-44 可知，电路的上限截止频率为 13.1kHz，下限截止频率为 20.4Hz。该下限截止频率与计算值吻合得很好，但正如 6.8.2 节中所述，上限截止频率较目标频率 20kHz 偏低。

逐渐减小电容 C_6 的取值，观察电路的上限截止频率。当 $C_6 = 36\text{pF}$ （系列值）时，电路的通频带如图 6-45 所示。此时，电路的上限截止频率为 19.5kHz，基本符合要求。

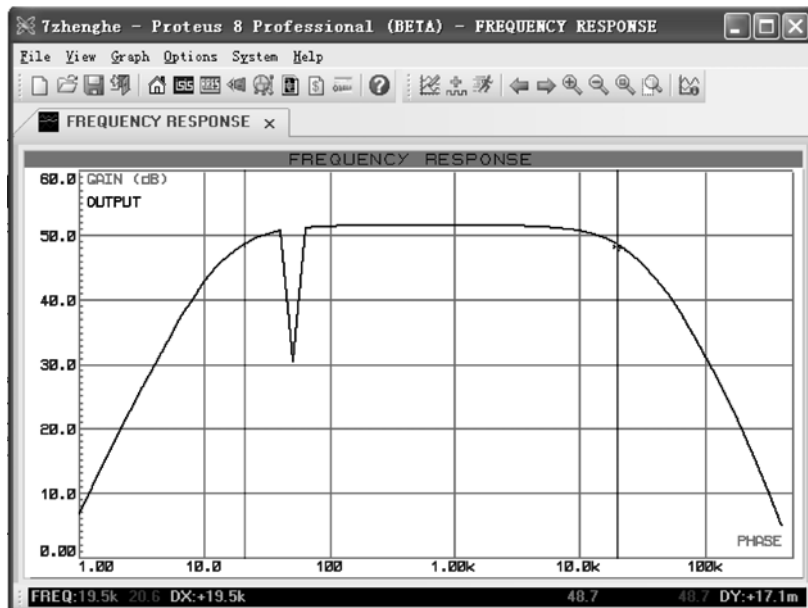


图 6-45 当 $C_6 = 36\text{pF}$ （系列值）时音频功放的通频带

6.8.4 电路整体的计算机仿真分析与验证

对电路进行噪声分析（不含音调控制电路），结果如图 6-46 和图 6-47 所示。

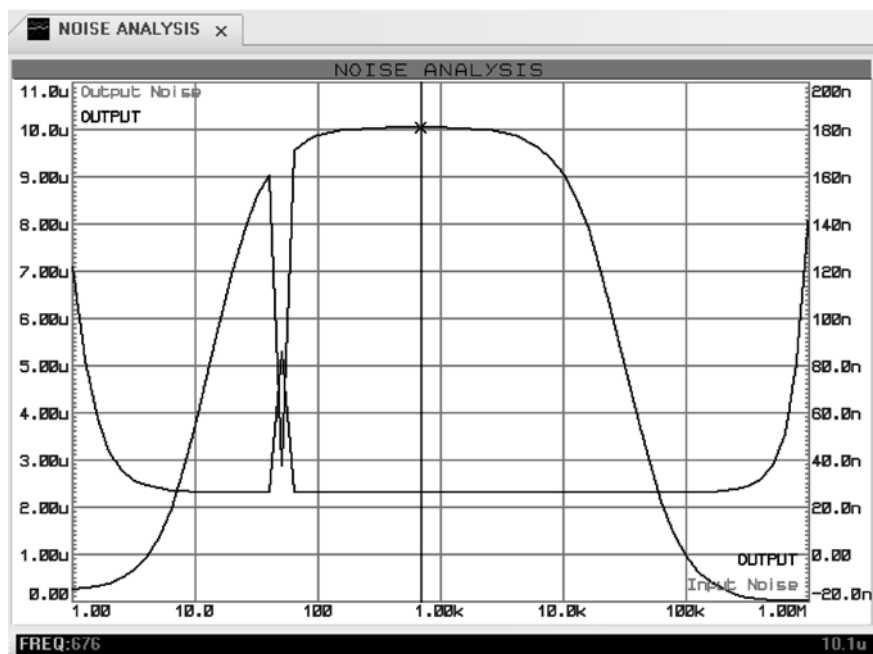


图 6-46 音频功放噪声分析（最大噪声电压）

由图 6-46 可以看出，最大噪声电压为 $10.1 \mu\text{V}/\sqrt{\text{Hz}}$ 。

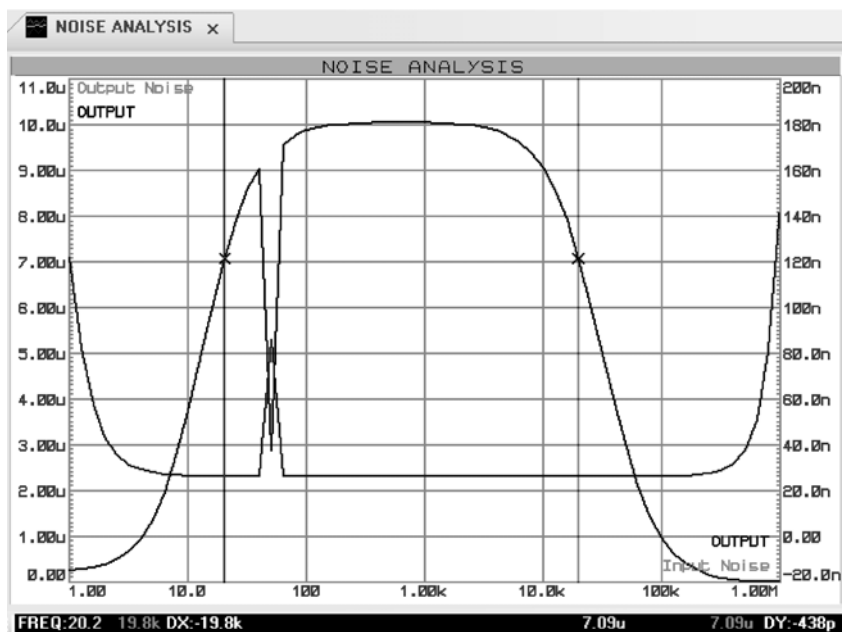


图 6-47 音频功放噪声分析（工作频率范围内的最小噪声电压）

由图 6-47 可知, 在电路的工作频率范围内, 输出电压噪声范围为 $6.54 \sim 9.32 \mu\text{V}/\sqrt{\text{Hz}}$ 。接入音调控制电路后, 对电路进行频率分析, 结果如图 6-48 所示。

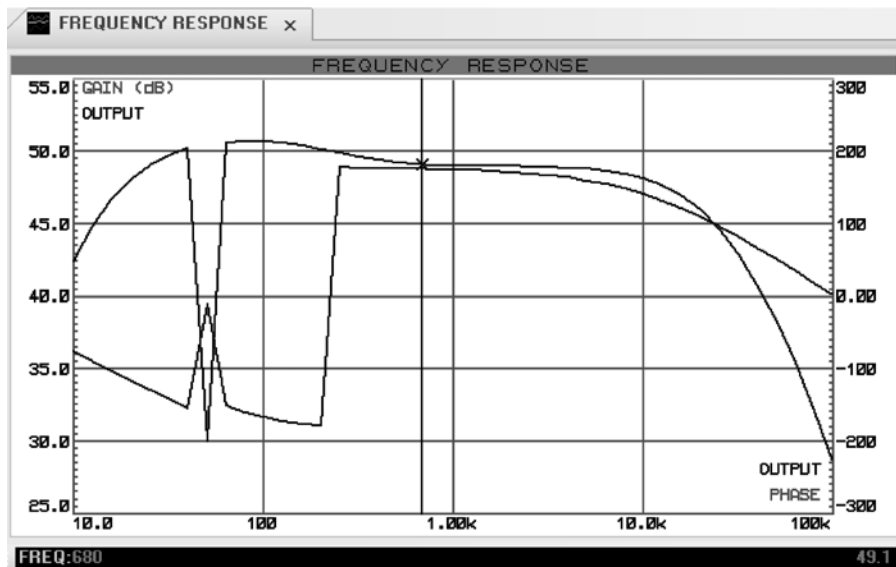


图 6-48 接入音调控制电路后电路的频率特性 (高低音电位器滑动端都在中点)

根据电路的频响特性, 能够预测音色的特征。比如, 当高频段提升的幅度比较大时, 声音听起来应比较尖锐。按此思路进行音频分析, 比较各种频响特性下的音质, 会得出与理论分析一致的结果。在音频信号源中添加钢琴曲 “Kiss the rain”, 钢琴曲的频率范围为 $26.5 \sim 4\,860\text{Hz}$, 通过调节低、高音控制电位器 RV1、RV2, 使上限截止频率为 5kHz , 如图 6-49 所示, 可以得到较为良好的音质, 如图 6-50 所示。

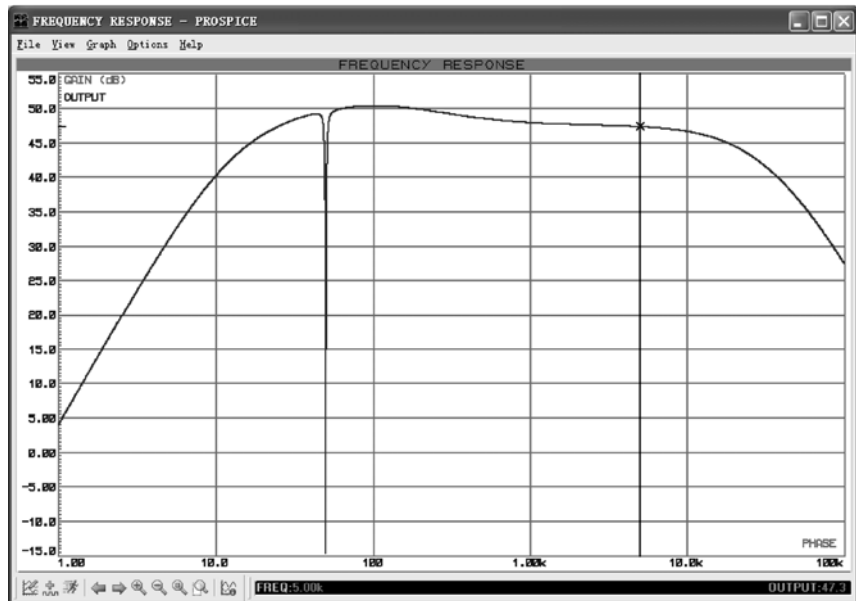


图 6-49 电路上限截止频率为 5kHz

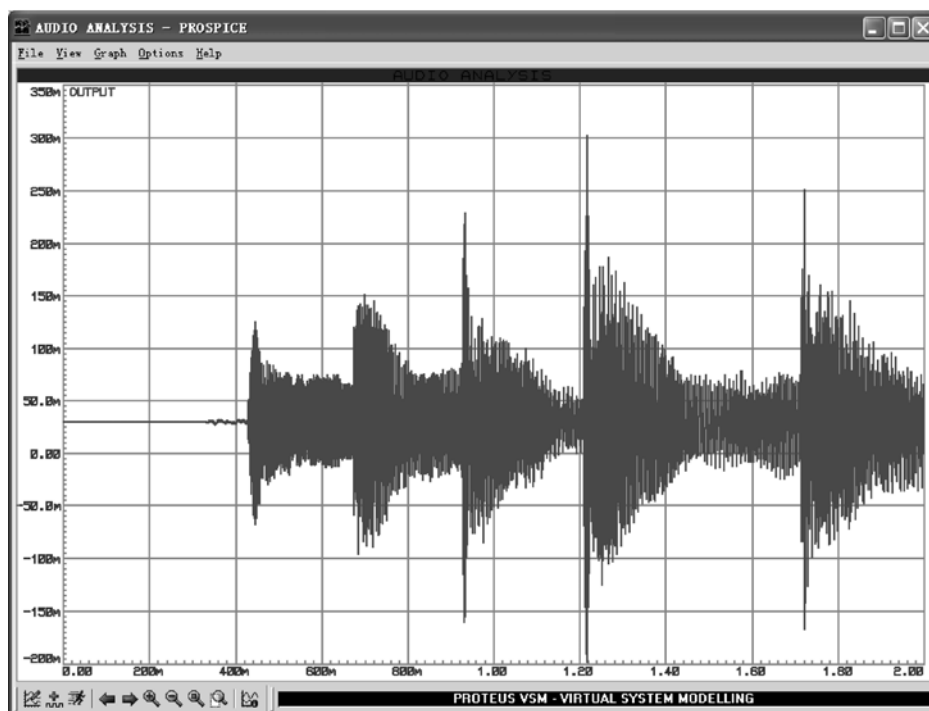
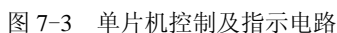


图 6-50 电路的音频分析

图 7-1 信号发生器总设计图



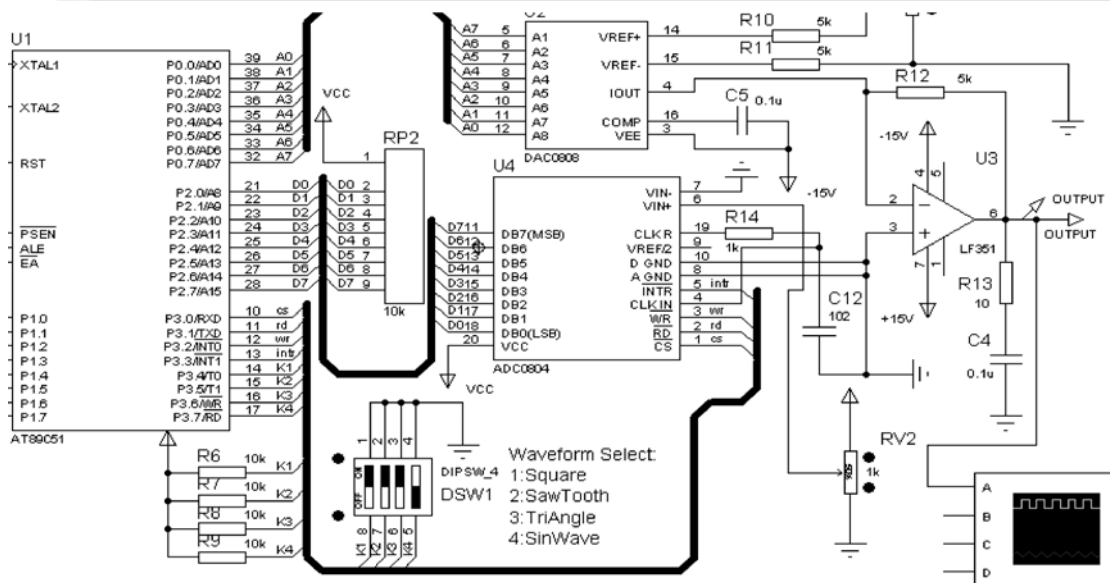


图 7-4 波形频率设定电路

【ADC0804 简介】ADC0804 是一种 8 位 COMS 依次逼近型的 A/D 转换器，三态锁定输出，存取时间大约为 $135\mu\text{s}$ ，8 位分辨率，转换时间大约为 $100\mu\text{s}$ ，总误差为 $\pm 1\text{LSB}$ ，工作温度区间为 $0\sim 70^\circ\text{C}$ ，引脚图见图 7-5。

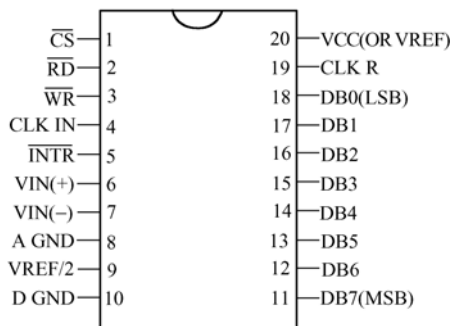


图 7-5 ADC0804 引脚图

- $\overline{\text{CS}}$ 芯片选择信号。
- $\overline{\text{RD}}$ 外部读取转换结果的控制输出信号。 $\overline{\text{RD}}$ 为 HI 时，DB0~DB7 处理高阻抗； $\overline{\text{RD}}$ 为 LO 时，数字数据才会输出。
- $\overline{\text{WR}}$ 用来启动转换的控制输入，相当于 ADC 的转换开始（ $\overline{\text{CS}}=0$ 时），当 $\overline{\text{WR}}$ 由 HI 变为 LO 时，转换器被清除；当 $\overline{\text{WR}}$ 回到 HI 时，转换正式开始。
- CLK IN, CLK R: 时钟输入或接振荡元件（R、C），频率限制在 $100\sim 1460\text{kHz}$ 。如果使用 RC 电路，则其振荡频率为 $1/(1.1RC)$ 。
- INTR: 中断请求信号输出，低电平动作。
- VIN(+), VIN(-): 差动模拟电压输入。输入单端正电压时，VIN(-)接地；而差动输入时，直接加入 VIN(+), VIN(-)。

- AGND、D GND: 模拟信号以及数字信号的接地。
- VREF: 辅助参考电压。
- DB0~DB7: 8 位的数字输出。
- VCC: 电源供应以及作为电路的参考电压。

ADC0804 参考原理图如图 7-6 所示。

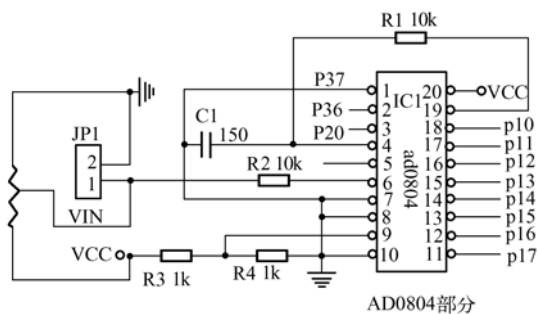


图 7-6 ADC0804 原理图

ADC0804 模数转换表见表 7-1。

表 7-1 ADC0804 模数转换表

十六进制	二进制	与满刻度的比率		相对电压值 $V_{REF}=2.560V$	
		高四位字节	低四位字节	高四位电压	低四位电压
F	1111	15/16	15/256	4.800	0.300
E	1110	14/16	14/256	4.480	0.280
D	1101	13/16	13/256	4.160	0.260
C	1100	12/16	12/256	3.840	0.240
B	1011	11/16	11/256	3.520	0.220
A	1010	10/16	10/256	3.200	0.200
9	1001	9/16	9/256	2.880	0.180
8	1000	8/16	8/256	2.560	0.160
7	0111	7/16	7/256	2.240	0.140
6	0110	6/16	6/256	1.920	0.120
5	0101	5/16	5/256	1.600	0.100
4	0100	4/16	4/256	1.280	0.080
3	0011	3/16	3/256	0.960	0.060
2	0010	2/16	2/256	0.640	0.040
1	0001	1/16	1/256	0.320	0.020
0	0000			0	0

1. 信号的产生

利用 8 位转换器 DAC0808, 可以将 8 位数字量转换成模拟量输出。数字量输入的范围为 0~255 之间, 对应的模拟量输出的范围在 V_{REF-} ~ V_{REF+} 之间。根据这一特性, 我们可以利用单片机的并行口输出的数字量, 产生常用的波形。

例如, 要产生幅度为 0~5V 的锯齿波, 只要将 DAC0808 的 V_{REF-} 接地, V_{REF+} 接 +5V, 单片机的并行口首先输出 00H, 再输出 01H、02H, 直到输出 FFH, 再输出 00H, 依次循环, 这样在图 7-1 所示的 VOUT 端就可以看到在 0~5V 之间变化的锯齿波。

2. 信号幅度控制

如上所述, DAC0808 的模拟量输出范围为 $V_{REF-} \sim V_{REF+}$, 也就是说, 当数字量输入为 00H 时, DAC0808 的输出为 V_{REF-} , 当输入为 FFH 时, DAC0808 的输出为 V_{REF+} 。所以, 为了调节输出波形的幅度, 只要调节 V_{REF} 即可。如图 7-1 所示, 在 V_{REF+} 端串接一电位器, 调节 V_{REF} 的电压, 即可达到调节波形幅度的目的。

3. 信号频率控制

仍以锯齿波为例, 若要调节信号的频率, 只需在单片机输出的两个数据之间加入一定的延时即可。如图 7-3 所示, 通过调节输入 ADC0804 转换的模拟电压值, 从而产生 8 位二进制数作为延时函数, 即可控制输出波形的幅值与频率。

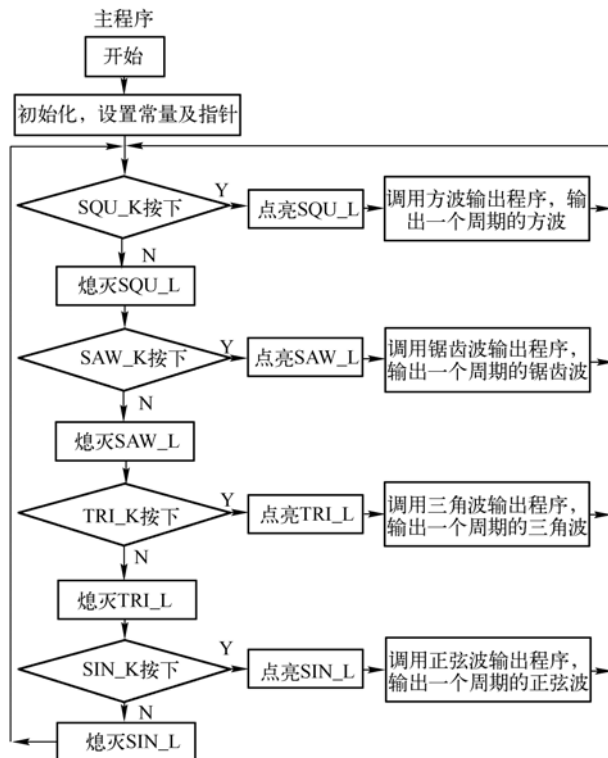
4. 波形切换

如图 7-4 所示, 利用 4 位 DIP 开关 DSW1 来选择波形, 并通过 4 个 LED 进行指示。



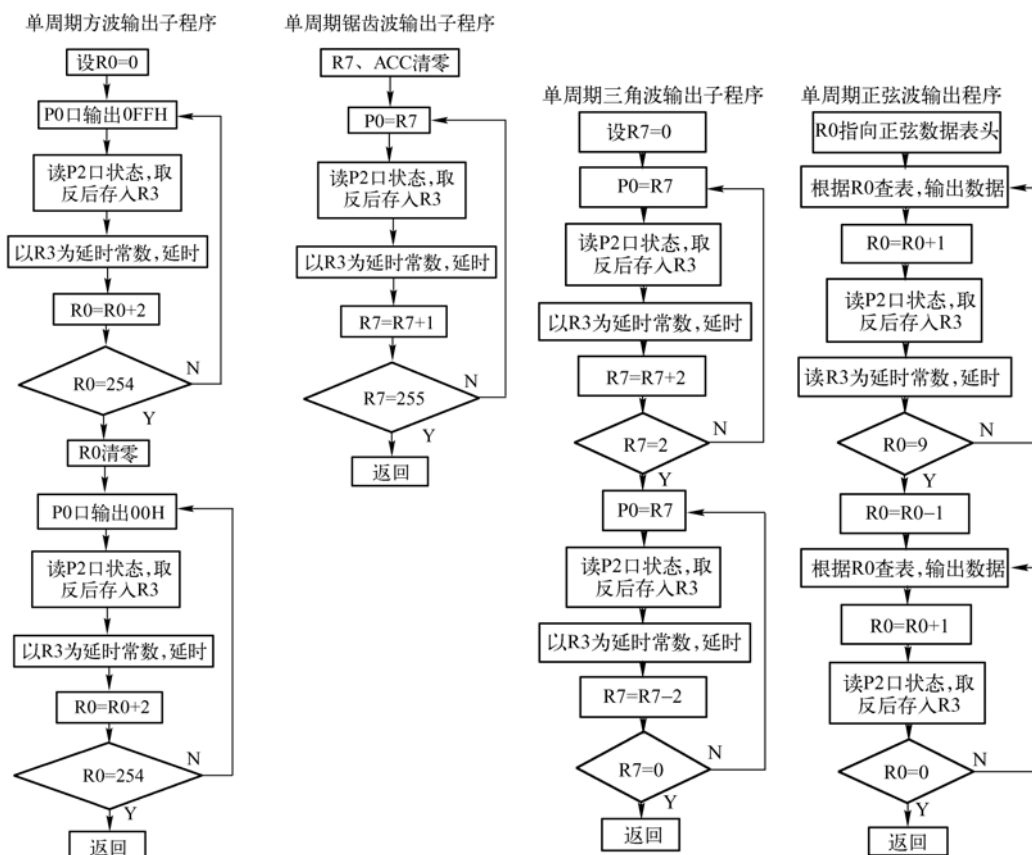
7.3 汇编语言程序设计流程

程序流程图如图 7-7 所示。



(a) 主程序

图 7-7 程序流程图



(b) 输出子程序

图 7-7 程序流程图 (续)



7.4 汇编语言程序源代码

```

ORG      00H
SQU_K BIT    P3.4
SAW_K BIT    P3.5
TRI_K BIT    P3.6
SIN_K BIT    P3.7

SQU_L BIT    P1.0
SAW_L BIT    P1.1
TRI_L BIT    P1.2
SIN_L BIT    P1.3

START:MOV     P1,#0FFH
        MOV     P2,#0FFH
        MOV     P3,#0FFH
        MOV     DPTR,#SIN_TAB
MAIN:MOV      P0,#00H
        JNB     SQU_K,S1

```

```

        SETB     SQU_L
        JNB      SAW_K,S2
        SETB     SAW_L
        JNB      TRI_K,S3
        SETB     TRI_L
        JNB      SIN_K,S4
        SETB     SIN_L
        SJMP     MAIN
S1:  CLR      SQU_L
      LCALL    SQUARE
      SJMP     MAIN
S2:  CLR      SAW_L
      LCALL    SAWTOOTH
      SJMP     MAIN
S3:  CLR      TRI_L
      LCALL    TRIANG
      SJMP     MAIN
S4:  CLR      SIN_L
      LCALL    SINWAVE
      SJMP     MAIN
SQUARE:
      MOV      R0,#00H
J11:  MOV      P0,#0FFH
      MOV      P2,#0FFH
      MOV      A,P2
      CPL      A
      MOV      R3,A
L11:  DEC      R3
      CJNE     R3,#255,L11
      INC      R0
      INC      R0
      CJNE     R0,#254,J11
      MOV      R0,#00H
J12:  MOV      P0,#00H
      MOV      P2,#0FFH
      MOV      A,P2
      CPL      A
      MOV      R3,A
L12:  DEC      R3
      CJNE     R3,#255,L12
      INC      R0
      INC      R0
      CJNE     R0,#254,J12
      MOV      R0,#00H
      RET
SAWTOOTH:
      CLR      A
      MOV      R7,A
J21:  MOV      P0,R7
      MOV      P2,#0FFH
      MOV      A,P2

```

```
CPL      A
MOV      R3,A
L21: DEC  R3
      CJNE R3,#255,L21
      INC  R7
      CJNE R7,#255,J21
      RET
TRIANG:
      MOV  R7,#00H
J31: MOV  P0,R7
      MOV  P2,#0FFH
      MOV  A,P2
      CPL  A
      MOV  R3,A
L31: DEC  R3
      CJNE R3,#255,L31
      INC  R7
      INC  R7
      CJNE R7,#254,J31
J32: MOV  P0,R7
      MOV  P2,#0FFH
      MOV  A,P2
      CPL  A
      MOV  R3,A
L32: DEC  R3
      CJNE R3,#255,L32
      DEC  R7
      DEC  R7
      CJNE R7,#00,J32
      RET
SINWAVE:
      MOV  R0,#00H
K41: MOV  A,R0
      MOVC A,@A+DPTR
      MOV  P0,A
      INC  R0
      MOV  P2,#0FFH
      MOV  A,P2
      CPL  A
      MOV  R3,A
L41: DEC  R3
      CJNE R3,#255,L41
      CJNE R0,#92,K41
K42: DEC  R0
      MOV  A,R0
      MOVC A,@A+DPTR
      MOV  P0,A
      MOV  P2,#0FFH
      MOV  A,P2
      CPL  A
      MOV  R3,A
```



```

238,240,242,244,246,247,249,250,
251,252,253,254,255,255 };

/*****
// 读 ADC0804 子程序
/*****
unsigned char adc0804( void )
{
    uchar dat,i;
    R_D=1;
    W_R=1;
    INTad=1;          //读 ADC0804 前准备
    P2=0xff;          //P2 全部置\准备
    CS=0;
    W_R=0;
    W_R=1;            //启动 ADC0804 开始测电压
    while(INT1==1);    //查询等待 A/D 转换完毕产生的 INT 信号
    R_D=0;            //开始读转换后数据
    i=i;              //无意义语句, 用于延时等待 ADC0804 读数完毕
    dat=P2;           //读出的数据赋予 addate
    R_D=1;
    CS=1;             //读数完毕
    return(dat);      //返回最后读出的数据
}

/*****
//方波发生函数
/*****
void square()
{
    uchar a,b;
    for(a=0;a<127;a++)
    {
        P0=0xff;
        P2=0xff;
        b=adc0804();
        b=~b;
        while(b--);    //调节相位, b 的变化越大, 相位变化越小
    }
    for(a=0;a<127;a++)
    {
        P0=0;
        P2=0xff;
        b=adc0804();
        b=~b;
        while(b--);
    }
}

/*****
//锯齿波发生函数
/*****
void sawtooth()
{

```

```

    uchar a,b;
    for(a=0;a<255;a++)
    {
        P0=a;
        P2=0xff;
        b=adc0804();
        b=~b;
        while(b--);
    }
}

//*****
// 三角波发生函数
//*****

void triang()
{
    uchar a,b;
    for(a=0;a<254;a=a+2)
    {
        P0=a;
        P2=0xff;
        b=adc0804();
        b=~b;
        while(b--);
    }
    for(a;a>1;a=a-2)
    {
        P0=a;
        P2=0xff;
        b=adc0804();
        b=~b;
        while(b--);
    }
}

//*****
//正弦波发生函数
//*****

void sinwave()
{
    uchar a,b;
    for(a=0;a<92;a++)
    {
        P0=sin_tab[a];
        P2=0xff;
        b=adc0804();
        b=~b;
        while(b--);
    }
    for(a=a-1;a>0;a--)
    {
        P0=sin_tab[a];

```

```

        P2=0xff;
        b=adc0804();
        b=~b;
        while(b--);
    }
}
/*****
//主函数
/*****
void main()
{
    P1=0xff;
    P2=0xff;
    P3=0xff;
    while(1)
    {
        P0=0;
        if(SQU_K==0)
        {
            SQU_L=0;
            square();
        }
        SQU_L=1;
        if(SAW_K==0)
        {
            SAW_L=0;
            sawtooth();
        }
        SAW_L=1;
        if(TRI_K==0)
        {
            TRI_L=0;
            triang();
        }
        TRI_L=1;
        if(SIN_K==0)
        {
            SIN_L=0;
            sinwave();
        }
        SIN_L=1;
    }
}

```



7.6 系统仿真

在 PROTEUS 中对输出波形进行波形分析, 选择图 7-1 中不同的开关, 调节图 7-1 中的频率控制电位器 RV2, 可以得到如图 7-8 所示发生的波形, 其中左侧一系列频率大于右侧一

列的频率,同时,调节图 7-1 中幅值电位器 RV1 可以改变输出波形的幅值。

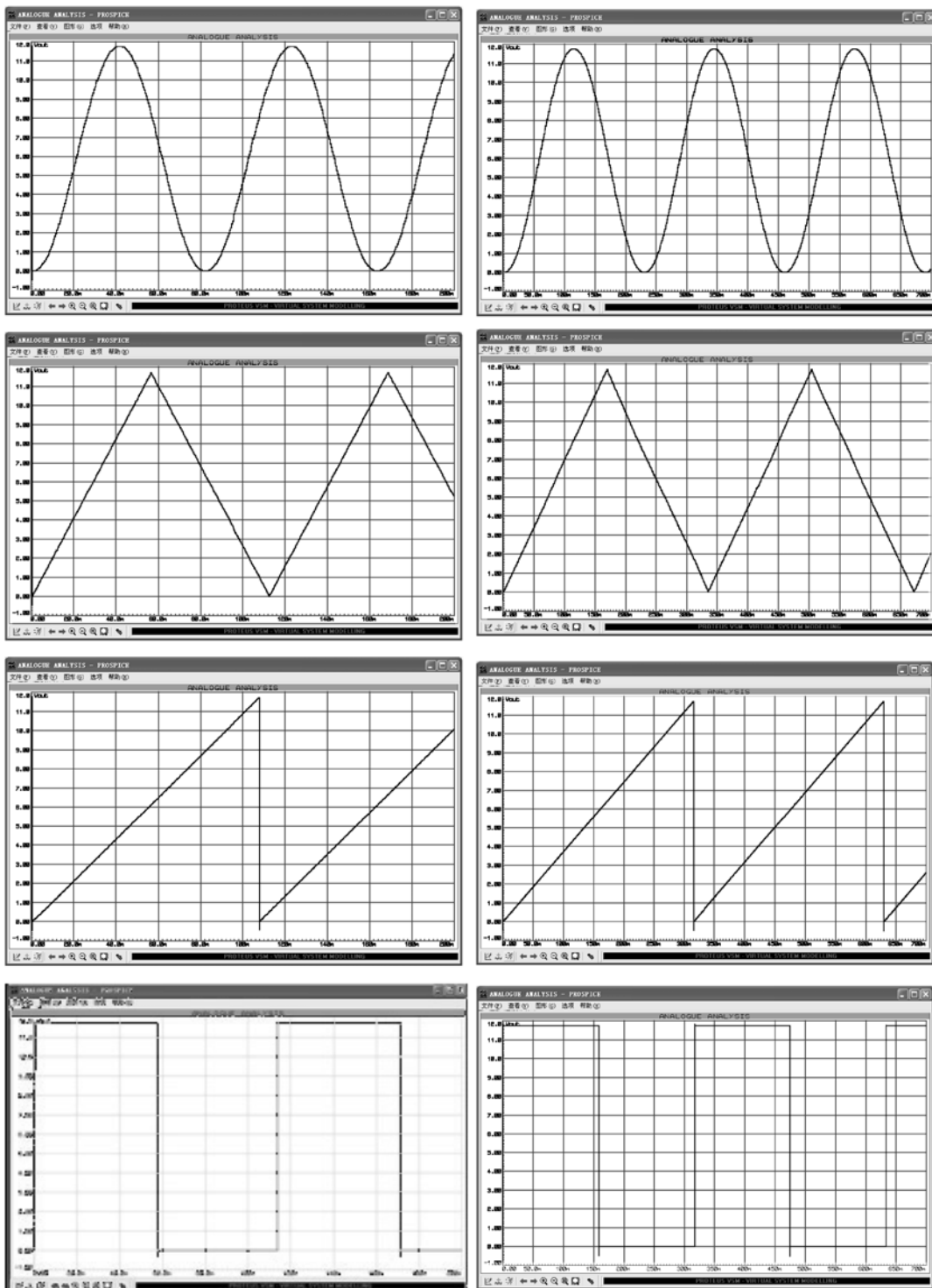


图 7-8 波形发生器输出波形



所以，在进行 PCB 设计之前需要添加两组连接器：电源输入端、信号输出端。另外，PCB 设计过程中不需要示波器，要在原理图中将其删除。

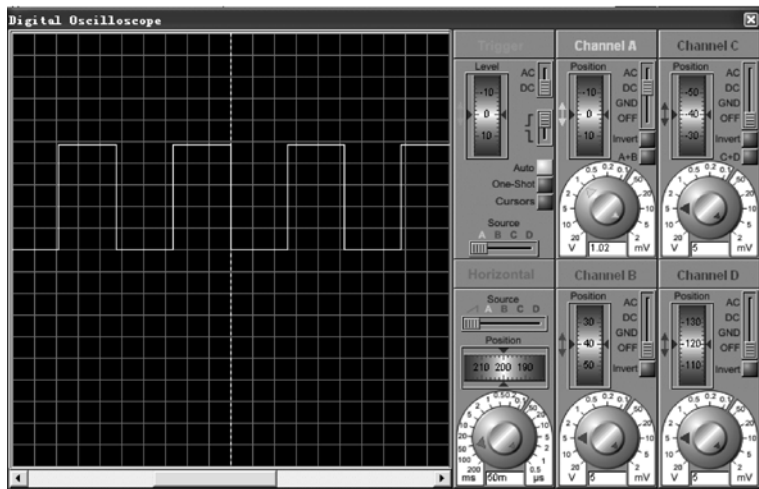



图 8-2 电路仿真结果

8.1.2 自定义元件符号

在原理图的后处理过程中，如果 ISIS 器件库中没有用户需要的元件，可根据需要自行绘制元件符号。所以以上面提及的两组连接器：电源输入端和信号输出端为例，说明在 PROTEUS ISIS 中绘制元件符号的方法。

【例 8-1】绘制 4 针连接器符号“POWER_CON_4P”，不定义封装。

单击 PROTEUS ISIS 工具箱中的 2D Graphics Box Mode 图标 ，在列表中选择 COMPONENT，在编辑区域中拖动鼠标左键，直至形成一个所需要的矩形框则可以松开鼠标左键，这时就绘制出一个矩形框，如图 8-3 所示。

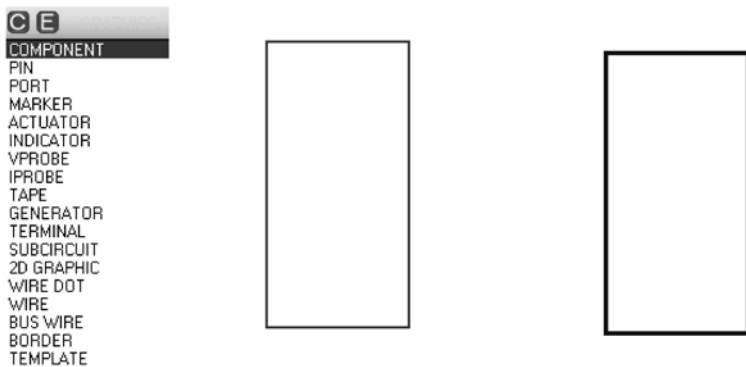



图 8-3 绘制矩形框

单击工具箱中的 Device Pins Mode 图标 ，则在列表中出现以下六种引脚类型，如图 8-4 所示。

- ☺ DEFAULT: 普通引脚;
- ☺ POSCLK: 上升沿有效的时钟输入引脚;
- ☺ SHORT: 较短引脚;
- ☺ INVERT: 低电平有效引脚;
- ☺ NEGCLK: 下降沿有效的时钟输入引脚;
- ☺ BUS: 总线。

首先用鼠标左键单击选择 DEFAULT 引脚类型，选中后单击下方的 Horizontal Reflection


图标, 将引脚水平翻转或者按小键盘上的“+”号翻转引脚; 在编辑窗口单击鼠标左键出现 DEFAULT 引脚, 按照图 8-5 所示开始添加 4 个引脚。在此处应注意添加引脚时的方向, 引脚中带有“×”号的一端为引脚的接线端, 要放在元件的外侧, 如图 8-5 所示。



图 8-4 六种类型的引脚

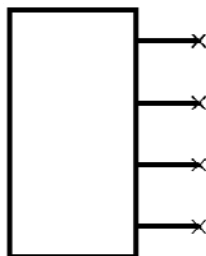
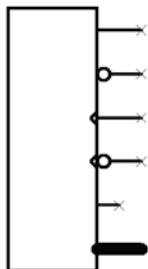


图 8-5 添加元件引脚 (DEFAULT 型)

然后开始添加引脚名及引脚号: 右击选中引脚, 再单击左键打开 Edit Pin 对话框, 在 Pin Name 栏中输入引脚名 P1, 在 Default Pin Number 栏中输入默认的引脚号 1, 如图 8-6 所示。

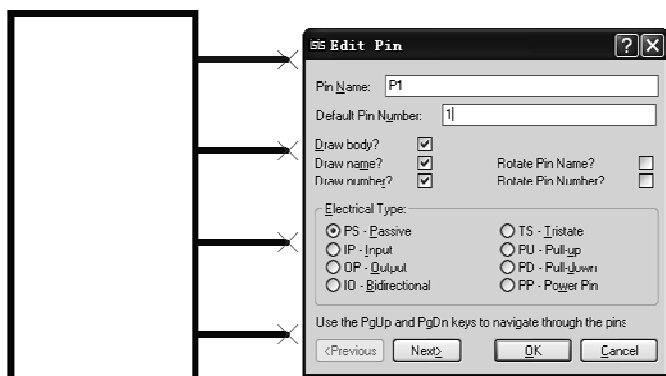
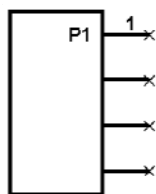


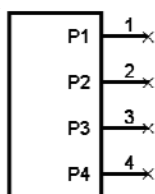
图 8-6 设置引脚属性

- | | |
|-----------------------------|-----------------------|
| ☺ PS-Passive: 无缘式的; | ☺ TS-Tristate: 三态引脚; |
| ☺ IP-Input: 输入引脚; | ☺ PU-Pull-up: 上拉引脚; |
| ☺ OP-Output: 输出引脚; | ☺ PD-Pull-down: 下拉引脚; |
| ☺ IO-Bidirectional: 双向作用引脚; | ☺ PP-Power Pin: 电源引脚。 |

按照图 8-6 所示设置其他选项, 设置完后单击“OK”按钮, 保存设置。此时选中的引脚设置如图 8-7 (a) 所示。按照图 8-7 (b) 所示编辑其他三个引脚的引脚名及引脚号。



(a)



(b)

图 8-7 编辑元件引脚

选中整个元件符号，在 PROTEUS 的菜单栏中选择 Library→Make Device（见图 8-8），弹出 Make Device 对话框，如图 8-9 所示。

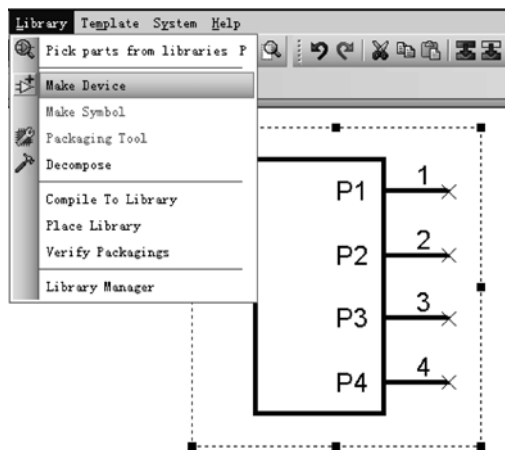


图 8-8 选择菜单栏选项

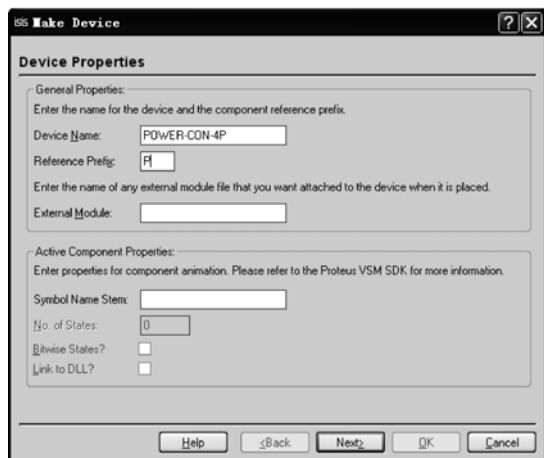


图 8-9 Make Device 对话框

在 General Properties 选项组中，设置 Device Name 为 POWER_CON_4P，在 Reference Prefix 栏中输入字母 P，单击“Next”按钮，进入下一步设置，定义元件封装，如图 8-10 所示。

如果暂时不能确定元件的封装情况，则可以跳过此步进行设置。单击“Next”按钮，进入下一步设置，设置元件属性，基本保持默认值即可，如图 8-11 所示。



图 8-10 定义元件封装

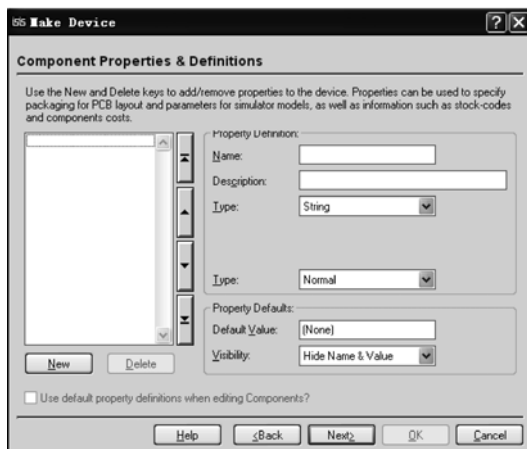


图 8-11 设置元件属性

单击“Next”按钮，定义元件的数据手册（Data Sheet），如图 8-12 所示。

单击“Next”按钮，设置元件索引，如图 8-13 所示。

- ☺ Device Category：元件所属类；
- ☺ Device Sub-category：元件所属子类；
- ☺ Device Manufacturer：元件制造厂商。



图 8-12 定义元素的数据手册

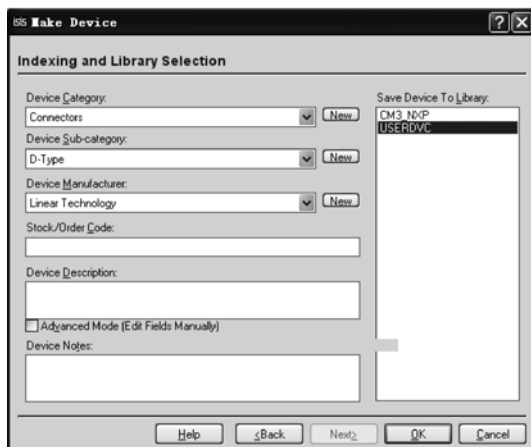


图 8-13 设置元素索引

单击“OK”按钮，完成元件定义，此时元件列表中自动添加了新建的元件 POWER_CON_4P，将其添加到原理图中的元件列表中，如图 8-14 所示。

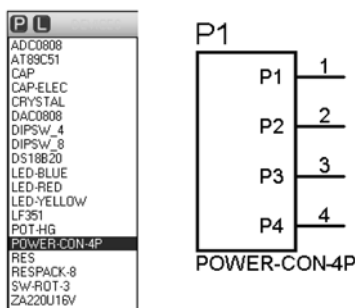



图 8-14 添加元件 POWER_CON_4P

【例 8-2】绘制 BNC 连接器符号 BND_1，并指定封装。

首先要按图 8-15 所示的模型绘制元件符号，单击 PROTEUS ISIS 工具箱中的 2D Graphics Circle Mode 图标，在列表中选择 COMPONENT，在编辑区域中拖动鼠标左键，直至形成一个所需要的圆形框则可以松开鼠标左键，这时就绘制出一个圆形框，如图 8-15 所示。

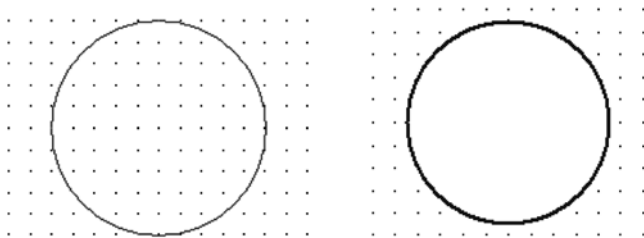




图 8-15 绘制圆形框

再单击列表选项中的 ACTUATOR 图标，在上述所画的圆形框中心位置单击鼠标左键，拖动鼠标左键，直至形成一个如图 8-16 所示的图形。

然后开始添加元件的引脚，按照图 8-17 所示开始添加两个引脚。单击工具箱中的 Device Pins Mode 图标，先用鼠标左键单击选择 DEFAULT 引脚类型，在编辑窗口中放置引脚 1，再點選 DEFAULT 引脚，选中后单击下方的 Rotate Anti-Clockwise 图标，将引脚逆时针进行翻转；在此处应注意添加引脚时的方向，引脚中带有“×”号的一端为引脚的接线端，要放在元件的外侧，如图 8-17 所示。

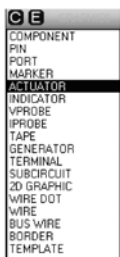


图 8-16 绘制双环圆形框

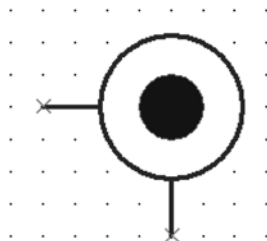
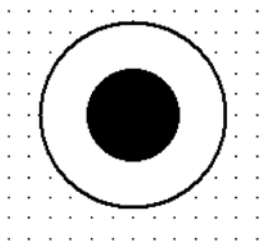


图 8-17 添加元件的引脚 (DEFAULT 型)

随后开始添加引脚名及引脚号：右击选中引脚，再单击左键打开 Edit Pin 对话框，在 Pin Name 栏中输入引脚名 P1，在 Default Pin Number 栏中输入默认的引脚号 1，如图 8-18 所示。

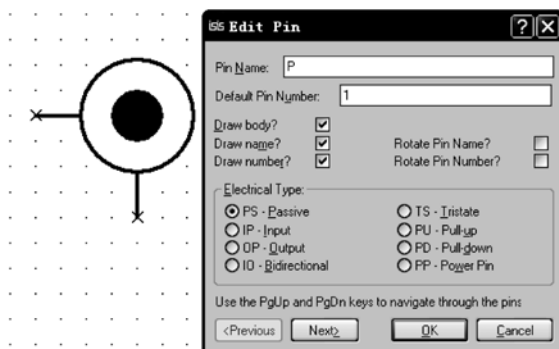
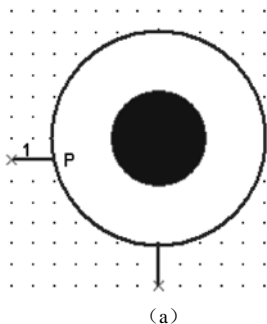
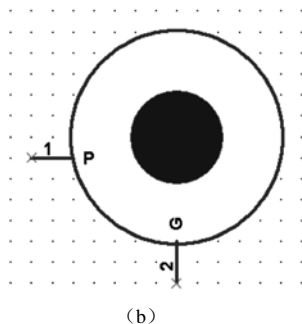


图 8-18 设置引脚属性

按照图 8-18 设置其他选项，设置完后单击“OK”按钮，保存设置。此时选中的引脚设置如图 8-19 (a) 所示。按照图 8-19 (b) 所示编辑另一个引脚的引脚名及引脚号。



(a)



(b)

图 8-19 编辑元件引脚

选中整个元件符号，在 PROTEUS 的菜单栏中选择 Library→Make Device（见图 8-8），弹出 Make Device 对话框，如图 8-20 所示。

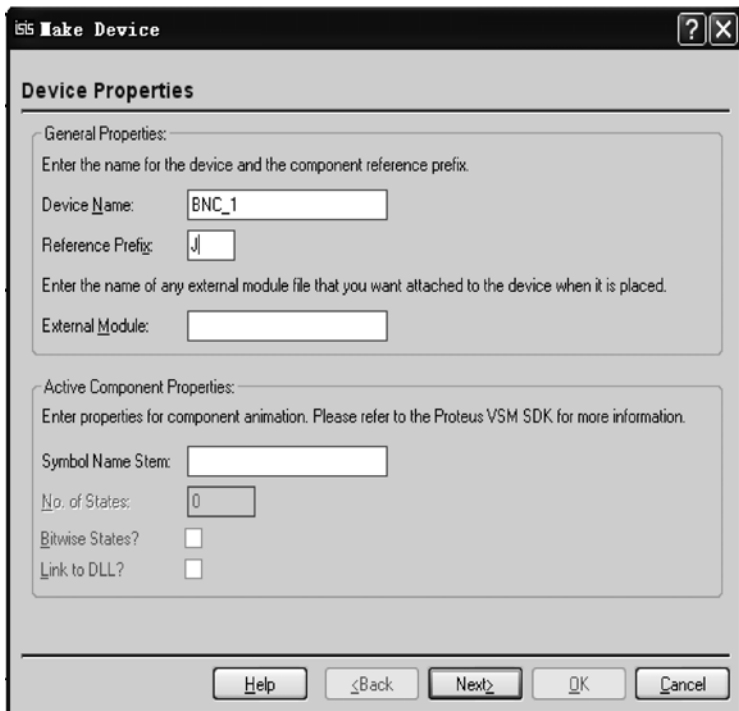


图 8-20 Make Device 对话框

单击“Next”按钮，进入下一步设置，如图 8-21 所示。单击“Add/Edit”按钮，打开 Package Device 对话框，如图 8-22 所示。

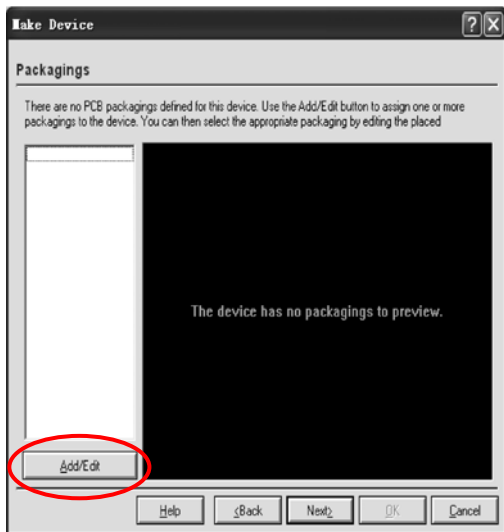


图 8-21 设置封装

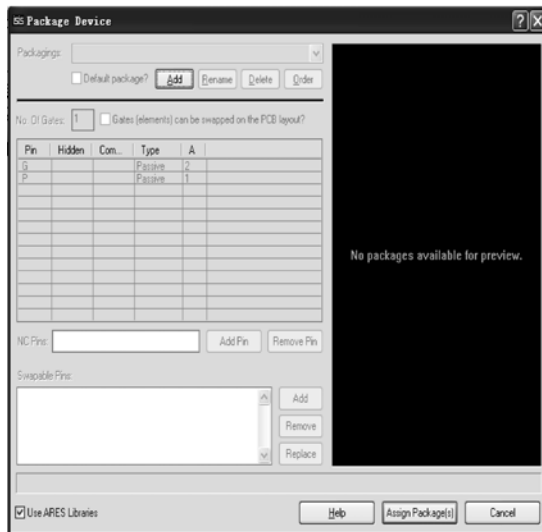


图 8-22 Package Device 对话框

单击“Add”按钮，选中 PROTEUS 库中自带的封装 RF-SMX-R，如图 8-23 所示。

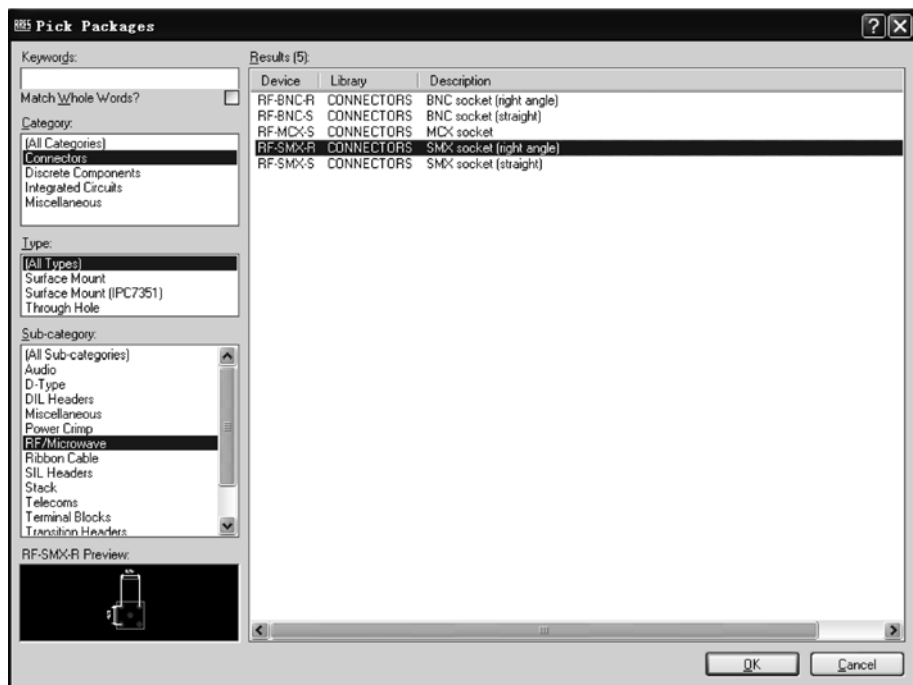


图 8-23 查找库中的封装

单击“OK”按钮，导入封装，如图 8-24 所示。

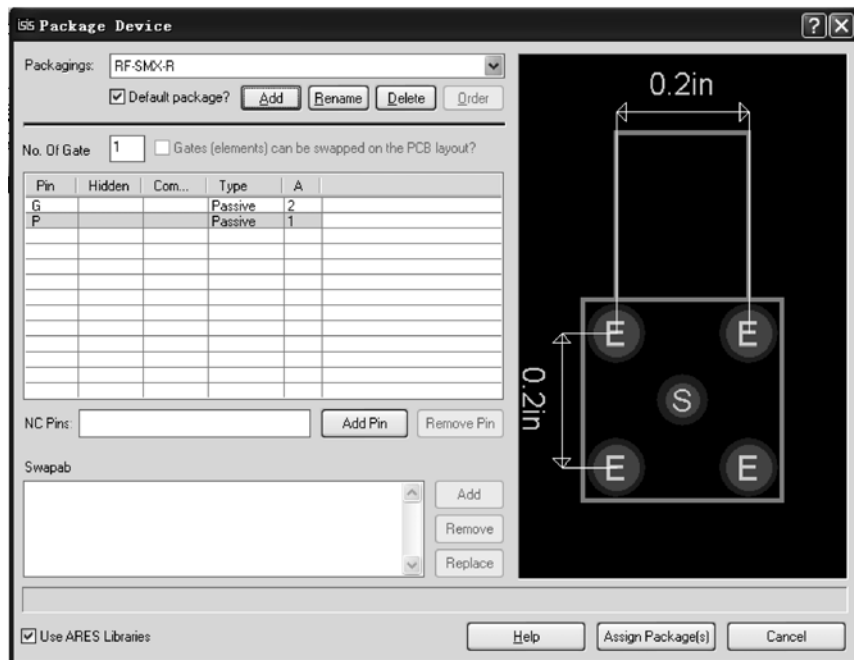


图 8-24 导入封装

在表格区中选中引脚号 1，在封装预览区中单击焊盘 S，这样就将元件符号中的 1 号引脚 P 映射为 PCB 封装中的引脚 S，如图 8-25 所示。同样，将 2 号引脚映射为焊盘 E。

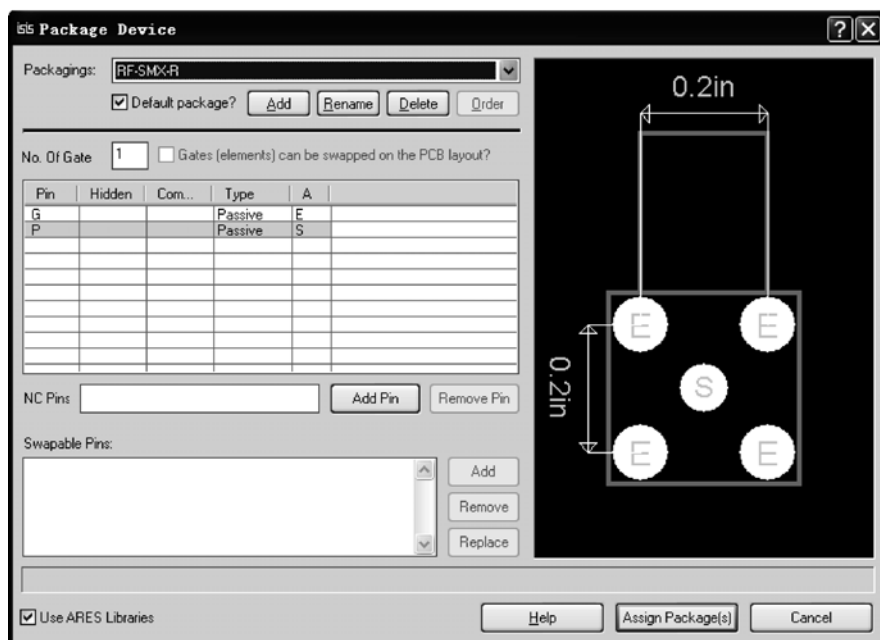


图 8-25 引脚映射

单击“Assign Package(s)”按钮，指定封装，如图 8-26 所示。

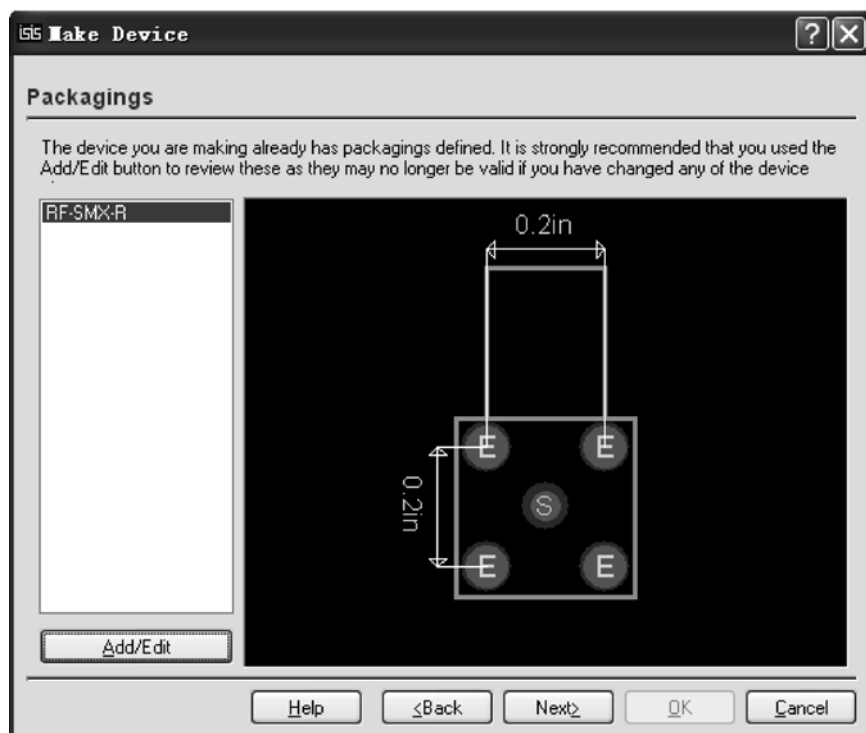


图 8-26 指定封装

单击“Next”按钮，定义元件属性，如图 8-27 所示。

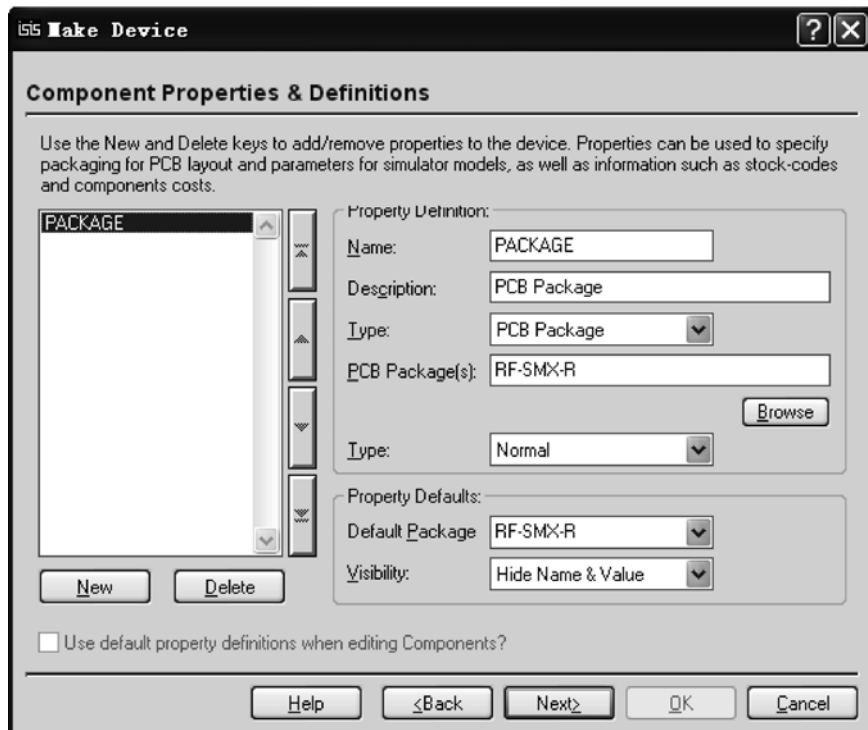


图 8-27 定义元件属性

单击“Next”按钮，定义器件手册，如图 8-28 所示。



图 8-28 定义器件手册

单击“Next”按钮，指定元件路径，如图 8-29 所示。

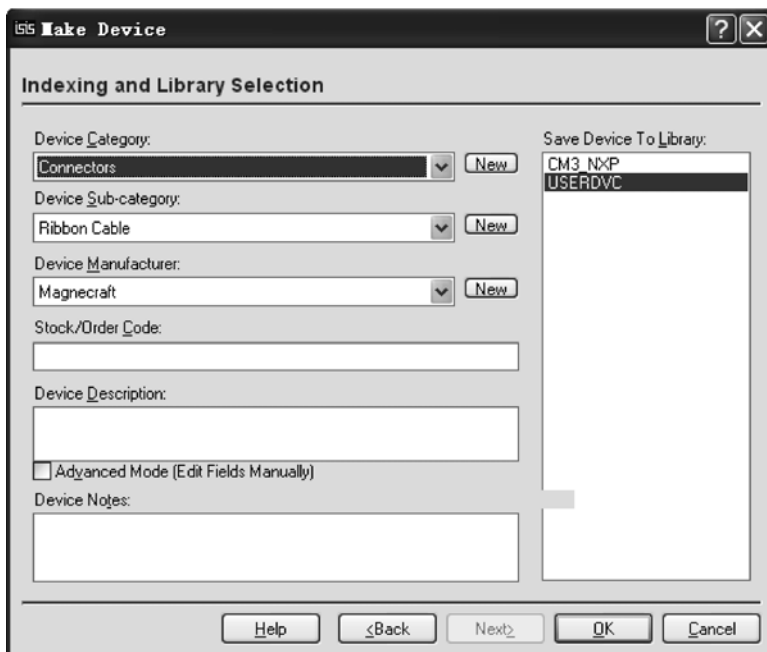


图 8-29 指定元件路径

单击“OK”按钮，即可完成元件符号制作。

8.1.3 检查元件的封装属性

右击选中一个元件符号 AT89C51，单击鼠标右键在列表中打开 Edit Component 对话框，选中窗口左下角的 Edit all properties as text 复选框，则所有元件属性都以文本显示，如图 8-30 所示。

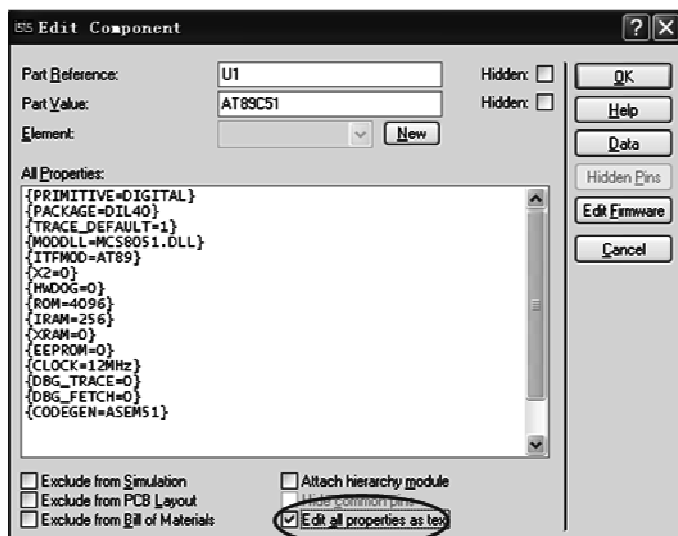


图 8-30 查看元件 AT89C51 的属性

此元件具有一条 PACKAGE=DIL40 属性, 说明此元件已被指定了封装, 封装名称为 DIL40。

右击选中元件符号 DIPSW_4, 在列表中单击鼠标左键打开 Edit Component 对话框, 查看元件属性, 可见此元件没有定义 PACKAGE 属性, 如图 8-31 所示。

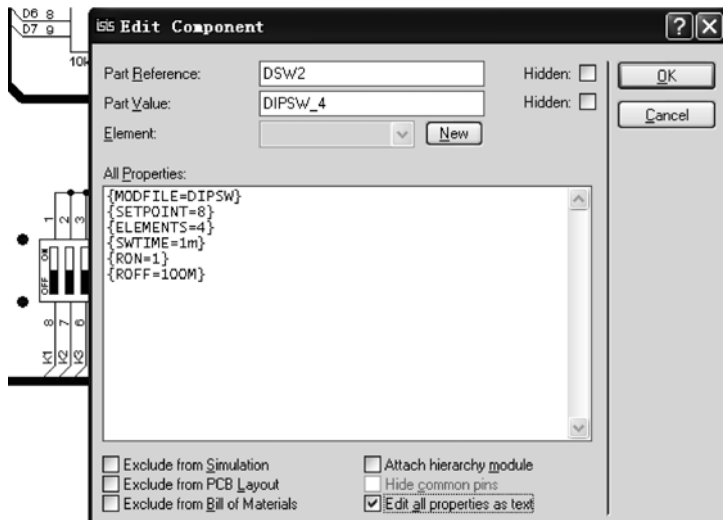


图 8-31 查看元件 DIPSW_4 的属性

对于没有 PACKAGE 属性的元件, 可以在 Edit Component 对话框为其添加 PACKAGE=?, 为其指定封装。现在在此对话框中输入 {PACKAGE=DIP_SW_4_8P}, 设置完成后如图 8-32 所示。

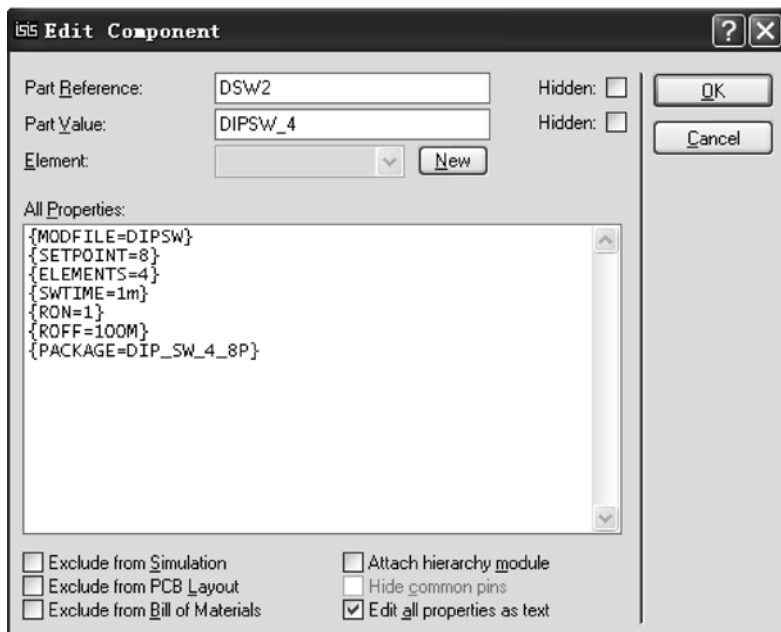


图 8-32 添加 PACKAGE

注意:

如果指定的封装是封装库里没有的,在进入 PCB 环境时会显示错误。用户可以在自己创建完成需要的元件封装后再指定。

8.1.4 完善原理图

在原理图中添加电源输入及信号输出的接口,如图 8-33 所示。

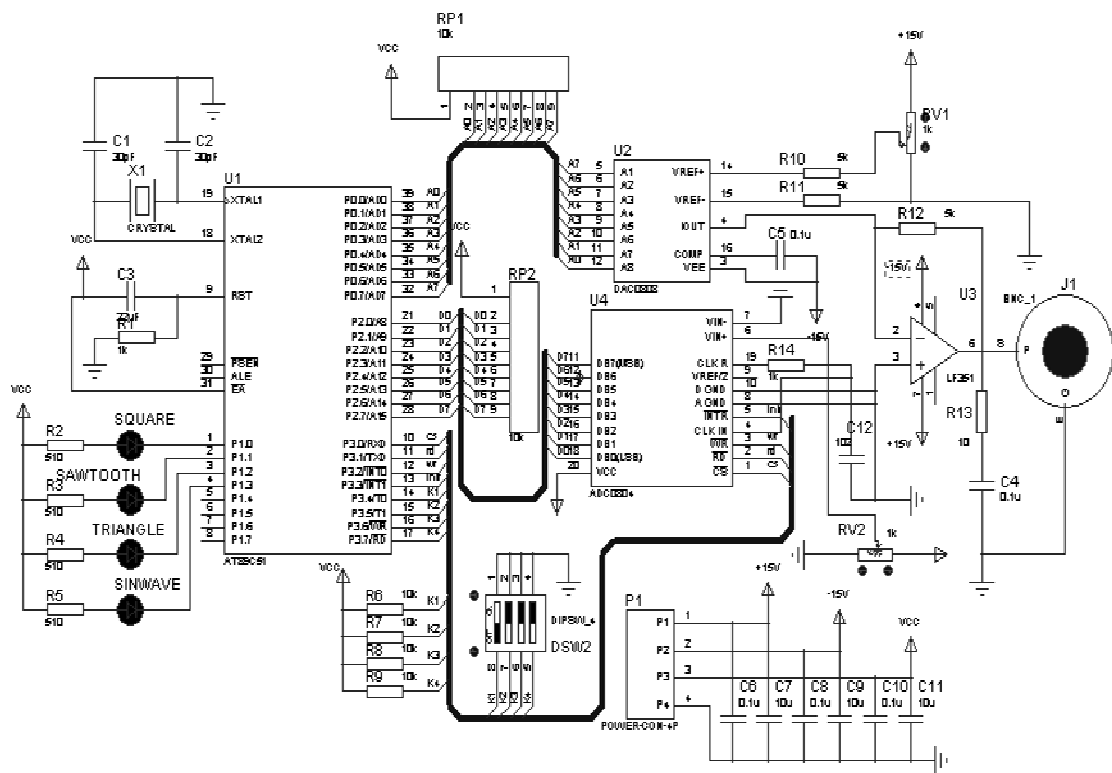


图 8-33 PCB 用原理图

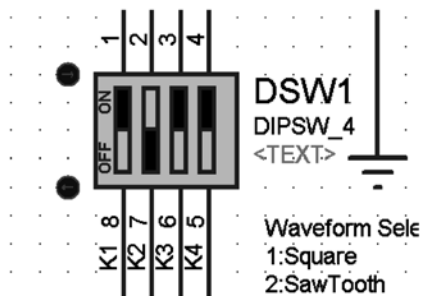


8.2 创建元件封装

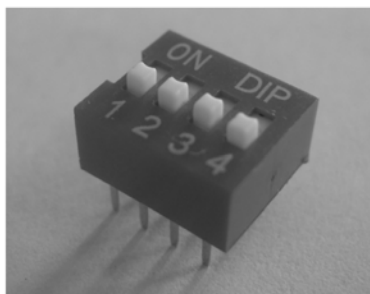
8.2.1 元件符号与元件封装

原理图中的元件符号反映的是元件的电气信息,包括网络及引脚之间的互连,引脚名与引脚号的对应关系等;而元件的封装反映的是元件的物理信息,包括元件外形、尺寸、引脚间距、引脚排列顺序等。下面举例说明:

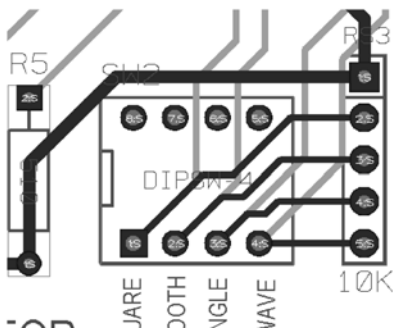
【例 8-3】DIP 开关 DIPSW_4 的符号、实物与 PCB 封装如图 8-34 所示。



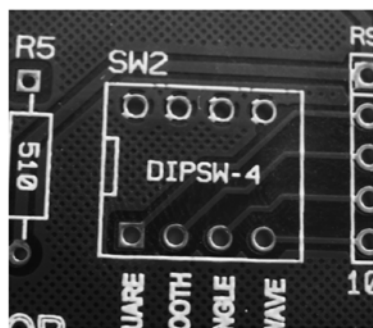
(a) 原理图符号



(b) 元件实物



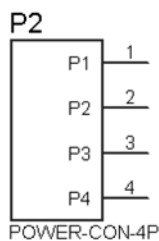
(c) PCB 封装



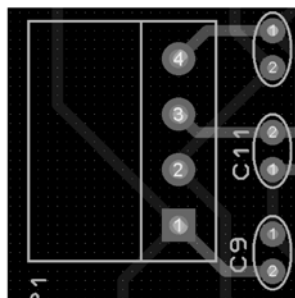
(d) 电路板

图 8-34 DIP 开关 DIPSW_4 的符号、实物与 PCB 封装

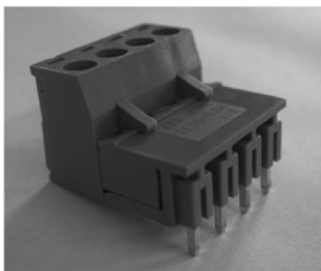
【例 8-4】电源插座 POWER_CON_4P 的符号、实物与 PCB 封装如图 8-35 所示。



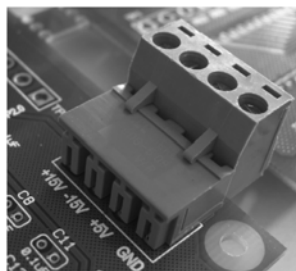
(a) 原理图符号



(b) PCB 封装



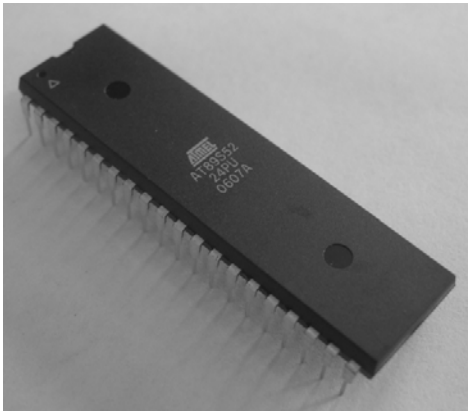
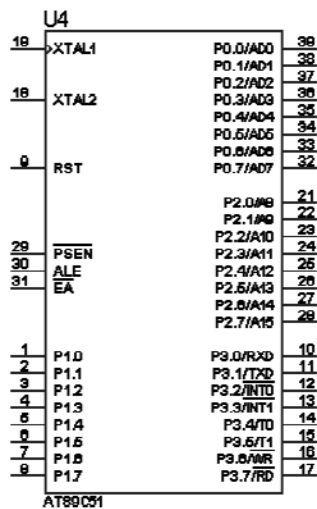
(c) 元件实物



(d) 电路板

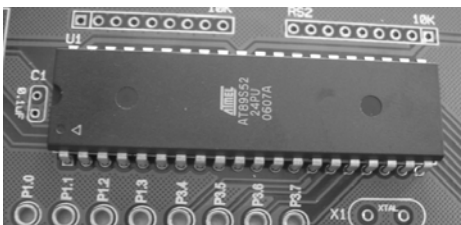
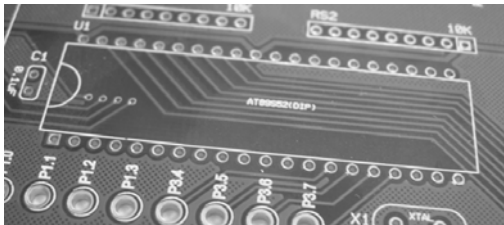
图 8-35 电源插座 POWER_CON_4P 的符号、实物与 PCB 封装

【例 8-5】单片机 AT89C51 的符号、实物与 PCB 封装如图 8-36 所示。

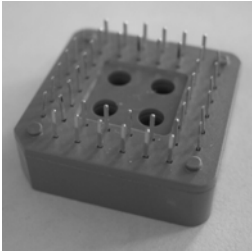
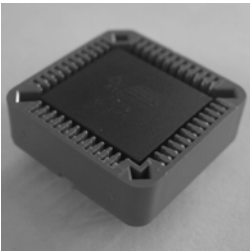
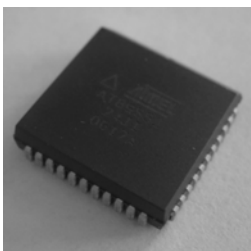


(a) 原理图符号

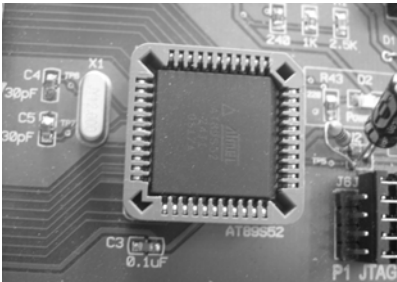
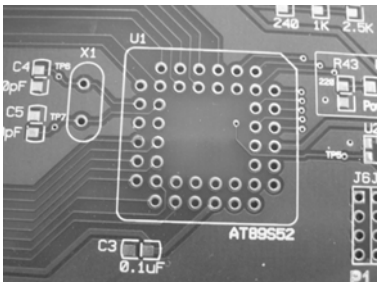
(b) 元件封装 1



(c) 电路板



(d) 元件封装 2



(e) 电路板


图 8-36 单片机 AT89C51 的符号、实物与 PCB 封装

8.2.2 创建元件封装

如果 PROTEUS 元件库中包含所需的封装，可以直接使用 PACKAGE 属性调用；如果没有，则需要预先创建元件封装。本节举例说明在 PROTEUS ARES 中创建元件封装的方法。

【例 8-6】制作 4 位拨码开关的封装 DIP_SW_4_8P。

在 PROTEUS 工具栏中单击 ，启动 ARES 界面，如图 8-37 所示。

单击 ARES 界面左侧工具栏中的 Square Through – hole Pad Mode 图标 ，在列表中选择焊盘 S-60-25，如图 8-38 所示。

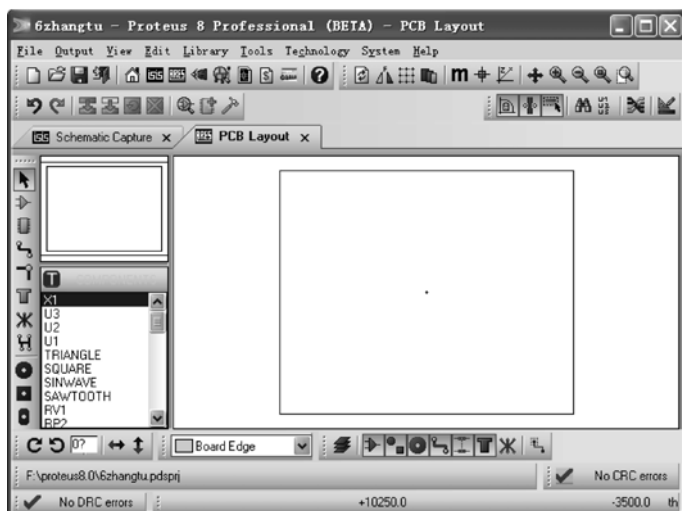


图 8-37 PROTEUS ARES 界面

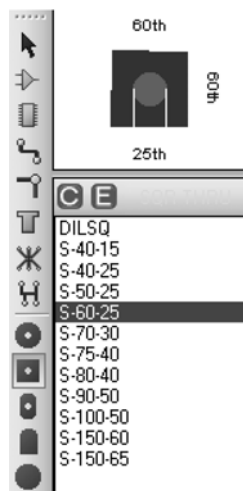


图 8-38 选择焊盘

在原点处单击鼠标左键，摆放选中的焊盘，并把它放在一个原点的位置上，如图 8-39 所示。

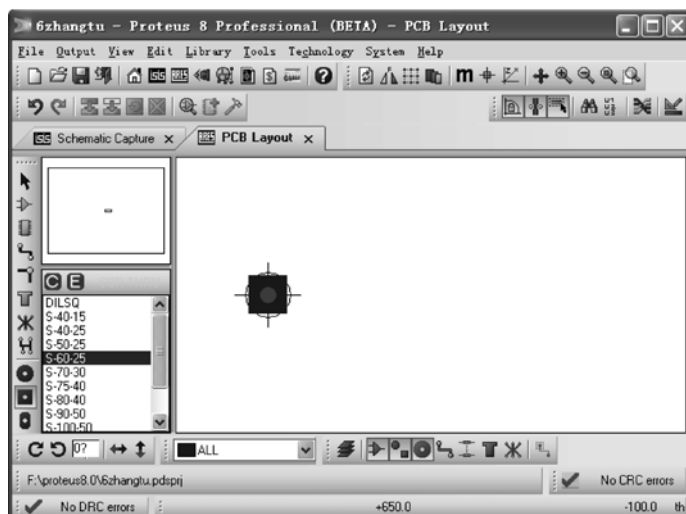




图 8-39 添加一个焊盘

在 ARES 窗口左侧工具箱中选择 Round Through – hole Pad Mode 图标, 此时需要在列表中选择焊盘 C-60-25, 而列表中没有所需要的焊盘, 则需要单击列表上方的 Create Pad Style 图标 (见图 8-40), 弹出 Create New Pad 对话框, 如图 8-41 所示。

在 Create New Pad 对话框的 Name 栏中输入焊盘名 C-60-25, 在 Normal 选项组中选中 Circular 选项, 单击“OK”按钮, 弹出 Edit Circular Pad 对话框, 如图 8-42 所示。

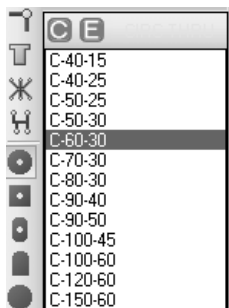


图 8-40 新建焊盘



图 8-41 Create New Pad 对话框



图 8-42 Edit Circular Pad 对话框

在 Edit Circular Pad 对话框中设置焊盘参数:

- ☺ Diameter (焊盘直径): 60th;
- ☺ Drill Mark (钻孔标记尺寸): 20th;
- ☺ Drill Hole (钻孔直径): 25th;
- ☺ Guard Gap (安全间距): 20th.

单击“OK”按钮, 完成焊盘设置, 此时焊盘列表中自动添加了新建的焊盘 C-60-25, 如图 8-43 所示。

选中焊盘 C-60-25, 在坐标 (100, 0) 处单击, 添加一个圆形焊盘 C-60-25, 如图 8-44 所示。

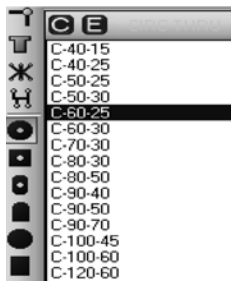


图 8-43 焊盘列表

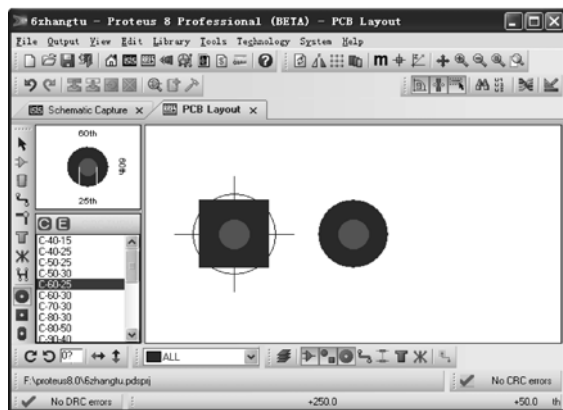


图 8-44 添加圆形焊盘 C-60-25

在编辑窗口右击选中新添加的圆形焊盘 C-60-25, 在菜单栏中选择 Edit→Replicate, 在弹出的 Replicate 对话框中设置复制的参数, 如图 8-45 所示。

单击“OK”按钮, 将选中的焊盘沿 X 轴方向复制两份, 间距为 100th, 如图 8-46 所示。

按上图选中 3 个圆形焊盘, 在菜单栏中选择 Edit→Replicate。在对话框中设置复制的

参数,如图 8-47 所示。

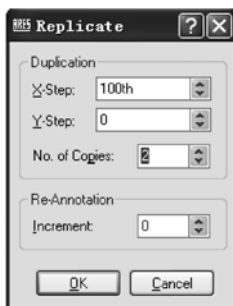


图 8-45 Replicate 对话框 1



图 8-46 批量复制焊盘 1



图 8-47 Replicate 对话框 2

单击“OK”按钮,则将把所选中的 7 个焊盘沿 Y 轴方向复制一份,间距为 400th,如图 8-48 所示。

在坐标 (0, 400) 处再添加一个圆形焊盘 C-60-25, 如图 8-49 所示。

右击选中左下角的方形焊盘,单击左键打开 Edit Single Pin 对话框,在 Number 栏中输入 1,如图 8-50 所示。

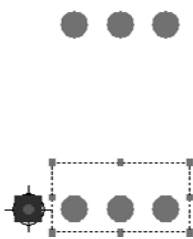


图 8-48 批量复制焊盘 2



图 8-49 添加焊盘

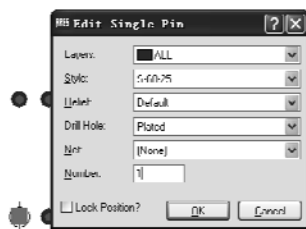




图 8-50 Edit Single Pin 对话框

单击“OK”按钮,确认并关闭对话框,此时焊盘上会显示引脚编号,如图 8-51 所示。

利用上述方法,按照图 8-52 所示为其余的引脚分配编号。

单击工具箱中的 2D Graphics Box Mode 图标 ,在左下方的下拉列表中选择层面 ,按照图 8-53 所示添加丝印外框。

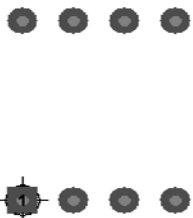


图 8-51 分配引脚编号 1

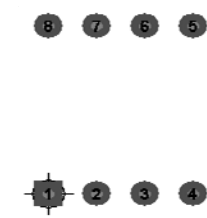


图 8-52 分配引脚编号 2

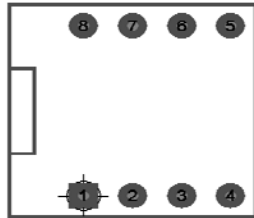



图 8-53 添加丝印外框

单击工具箱中的 2D Graphics Markers Mode 图标 ,在列表中选择 ORIGIN,在第一个焊盘处添加原点标记,如图 8-54 所示。

在选中的 2D Graphics Markers Mode 图标  的列表中选择 REFERENCE,如图 8-55 所示。然后再按图 8-56 所示添加元件 ID,添加 REFERENCE。

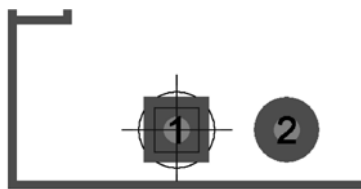


图 8-54 添加原点标记



图 8-55 在列表中选择 REFERENCE

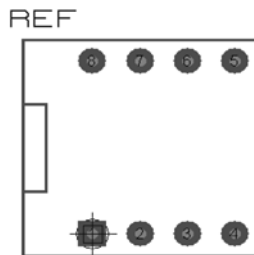


图 8-56 添加 REFERENCE

选中所有焊盘及丝印图形，在菜单栏中选择 Library→Make Package（见图 8-57），打开 Make Package 对话框，如图 8-58 所示。

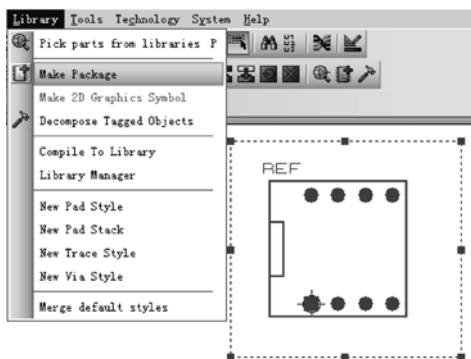


图 8-57 制作 Make Package

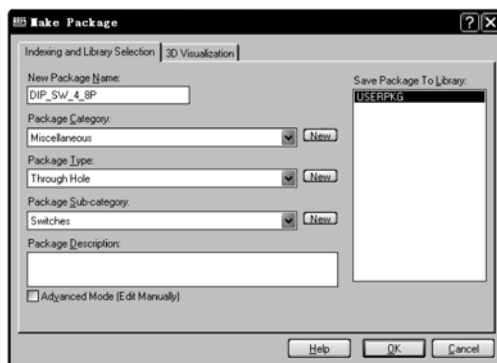


图 8-58 Make Package 对话框

☺ New Package Name: DIP_SW_4_8P;

☺ Package Category: Miscellaneous;

☺ Package Type: Through Hole;

☺ Package Sub-category: Switches。

单击“OK”按钮，保存封装。

利用上述方法可以制作其他元件封装，如图 8-59～图 8-61 所示。

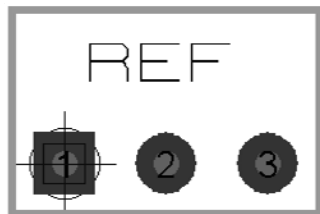


图 8-59 电位器封装 POT_HG_3P

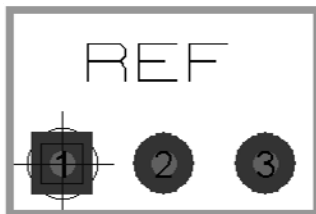


图 8-60 4 针电源插座封装 CON_4P_W200

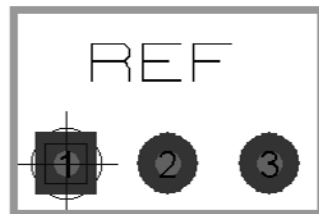


图 8-61 LED 封装 LED_100

在菜单栏中选择 File→Exit，退出 ARES 环境，返回 PROTEUS ISIS。

8.2.3 指定元件封装

在原理图中，右击选中一个命名为 SQUARE 的 LED，单击鼠标左键打开 Edit Component 对话框，选中窗口左下角的 Edit all properties as text 选项，在文本区域中添加

一条属性 {PACKAGE=LED_100}, 如图 8-62 所示。

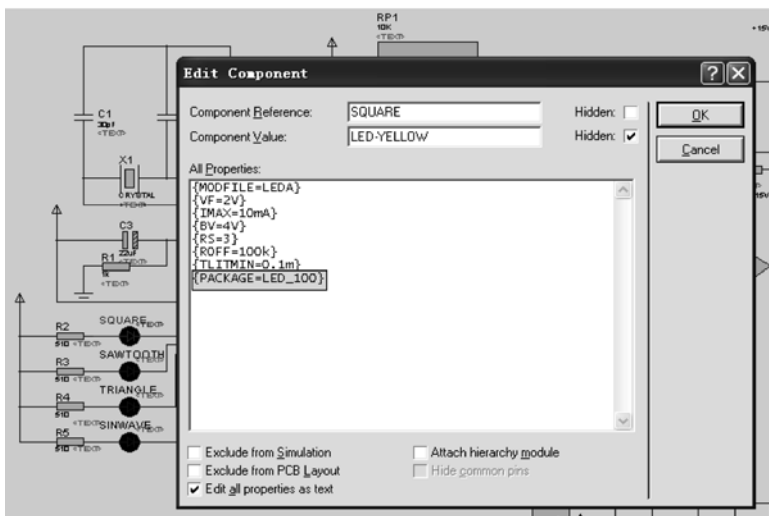


图 8-62 为 LED 指定封装

利用上述相同的方法, 为其他的三个 LED 指定封装为 PACKAGE=LED_100。

右击选中 4 位拨码开关, 单击鼠标左键打开 Edit Component 对话框, 选中窗口左下角的 Edit all properties as text 选项, 在文本区域中添加一条属性 {PACKAGE=DIP_SW_4_8P}, 如图 8-63 所示。

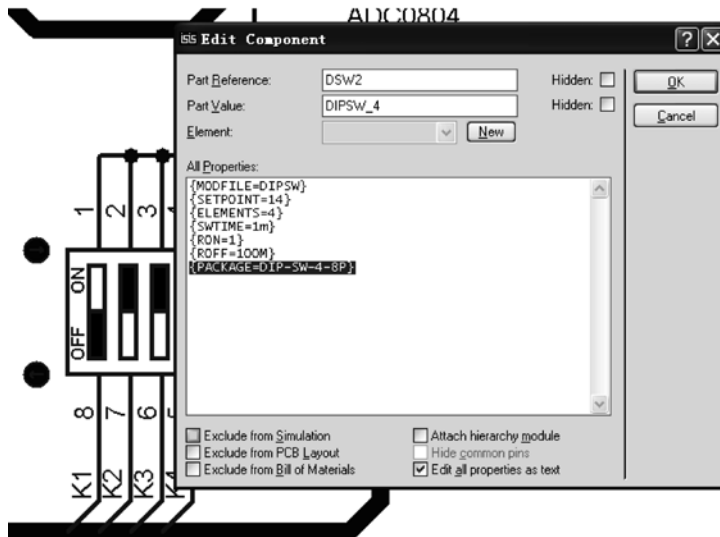



图 8-63 为 4 位拨码开关指定封装

为其他元件选择封装: POWER_CON_4P 选择封装 CON_4P_W200, POT-HG 选择封装 POT_HG_3P, LED-YELLOW 选择封装 LED_100。

在进行 PCB 设计之前应该先检查一下元件的封装是否全部指定或是否全部正确。单击工具栏图标 , 出现如图 8-64 所示窗口, 可以观察元件封装是否全部指定。

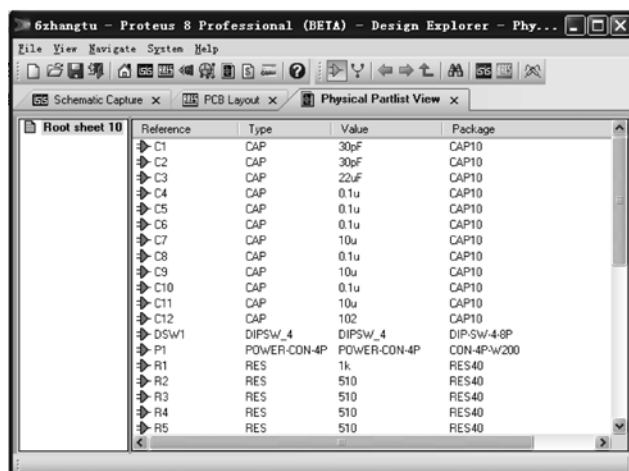


图 8-64 查看元件封装

在菜单栏选择 Library→Verify Packagings, 如图 8-65 所示, 查看元件封装是否有错误。如果没有错误, 则显示如图 8-66 所示提示框。

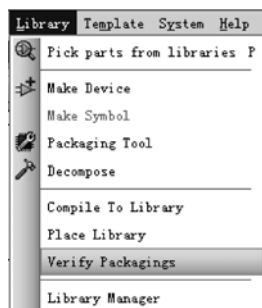


图 8-65 检查元件封装



图 8-66 检查无错误

单击工具栏图标 , 进入 ARES PCB 编辑环境, 如图 8-67 所示。

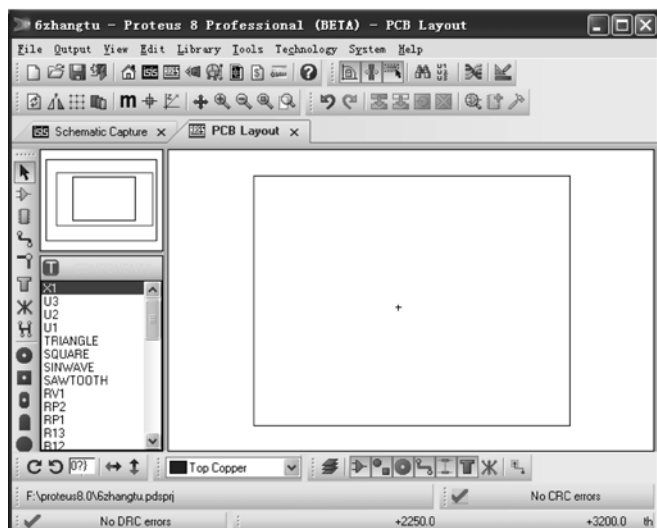


图 8-67 ARES PCB 编辑环境



8.3 PCB 布局

8.3.1 设置层面

进入 PROTEUS ARES 界面后, 在菜单栏中选择 Technology→Set Layer Usage, 弹出 Set Layer Usage 对话框, 具体设置如图 8-68 所示。

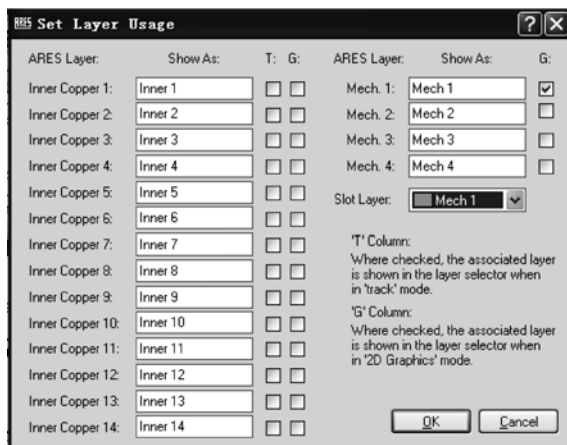


图 8-68 设置层面

8.3.2 自动布局

自动布局之前必须先画好板框, 可以先画一个大概的板框, 布局后根据实际大小再进行调整。


单击 ARES 界面左侧工具箱中的 2D Graphics 图标, 在左下角的层面列表中选择 Board Edge (见图 8-69), 绘制板框, 如图 8-70 所示。



图 8-69 选择图层

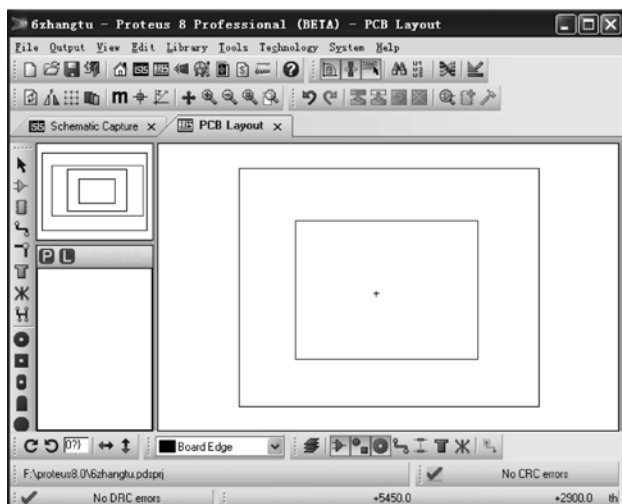


图 8-70 绘制板框

在菜单栏中选择 Tools→Auto Placer，打开 Auto Placer 对话框，如图 8-71 所示。

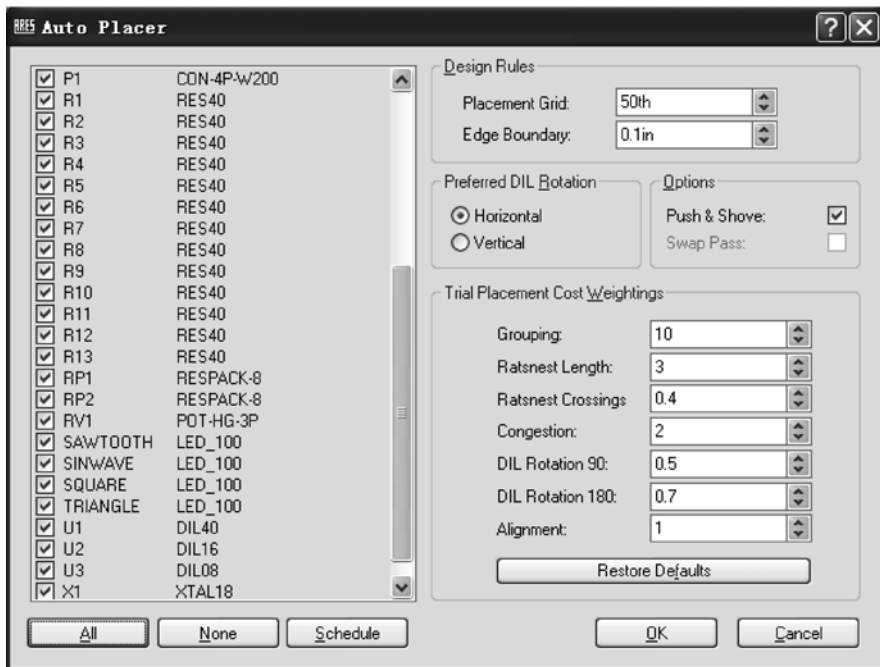


图 8-71 Auto Placer 对话框

- ☺ Design Rules: 设计规则。
 - ☞ Placement Grid: 布局格点;
 - ☞ Edge Boundary: 元件距电路板边框的距离。
- ☺ Preferred DIL Rotation: 元件的方向。
 - ☞ Horizontal: 水平;
 - ☞ Vertical: 垂直。
- ☺ Options: 选项。
 - ☞ Push & Shove: 推挤元件;
 - ☞ Swap Parts: 元件交换。
- ☺ Trial Placement Cost Weightings: 尝试摆放的权值。
 - ☞ Grouping: 群组;
 - ☞ Ratsnest Length: 飞线长度;
 - ☞ Ratsnest Crossings: 飞线交叉;
 - ☞ Congestion: 密集度;
 - ☞ DIL Rotation 90: 元件旋转 90° ;
 - ☞ DIL Rotation 180: 元件旋转 180° ;
 - ☞ Alignment: 对齐;
 - ☞ Restore Defaults: 恢复默认值。

在 Auto Placer 对话框的元件列表中选中所有元件，单击“OK”按钮，元件会逐个摆放到板框中，如图 8-72 所示。

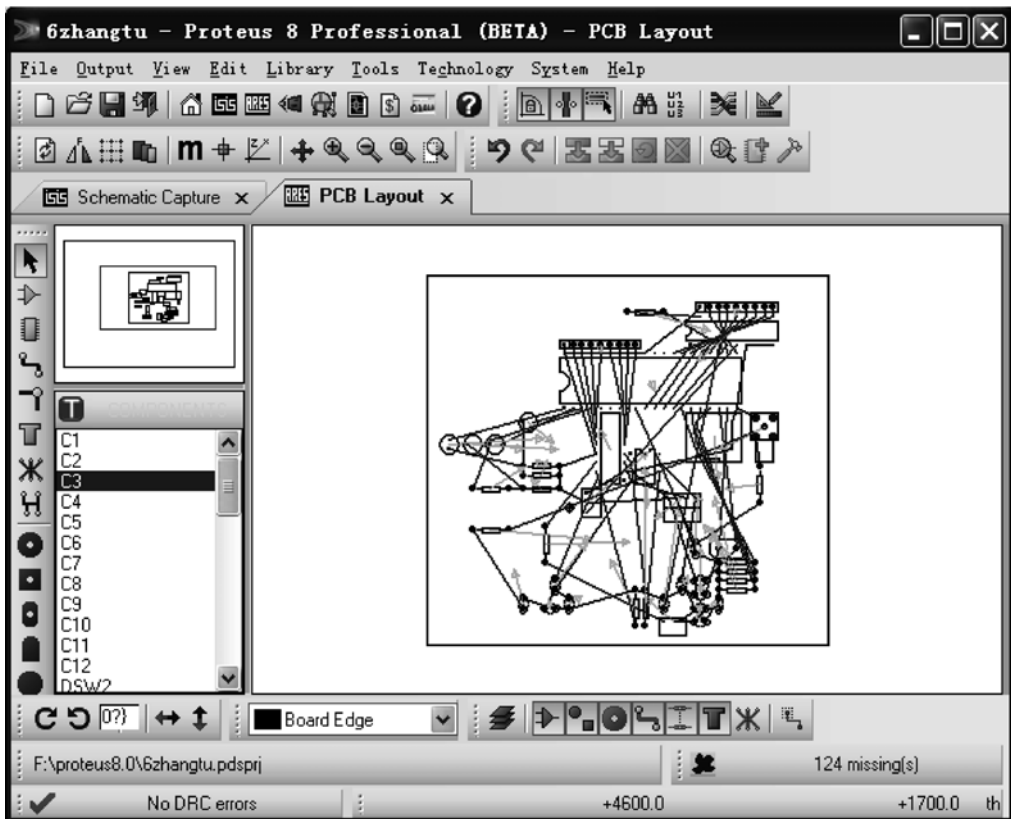



图 8-72 自动布局






注意:

在没有连线之前，会显示错误。

8.3.3 手工布局

单击 ARES 界面左侧工具箱中的 Component placement and editing 图标，在元件列表中会列出所有未摆放的元件。在列表中选中元件，在板框中单击鼠标左键，摆放选中的元件。

自动布局后手工调整或手工布局时用到的一些操作如下：

- ☺ 右键选中元件，拖到预期位置。选中的同时可按“+”键或“-”键旋转元件。
- ☺ 鼠标光标放在任意引脚上时，ARES 界面底部的状态栏将显示此引脚的属性。
- ☺ 按下 Edit Objects 按钮后，可直接单击元件，编辑其属性，相当于右击选中后，单击左键编辑属性。
- ☺ 在 PCB 的当前层垂直或按角度旋转。
- ☺ 对元件进行水平或垂直翻转。
- ☺ 对元件进行复制、移动、旋转和删除操作。
- ☺ 显示飞线和向量符号：在菜单栏中选择 View→Edit Layers Colours/Visibility，或者单击工具栏中的图标，弹出 Display Settings 对话框，如图 8-73 所示。选中 Ratsnest，显示飞线；取消选中 Force Vectors，不显示向量符号。

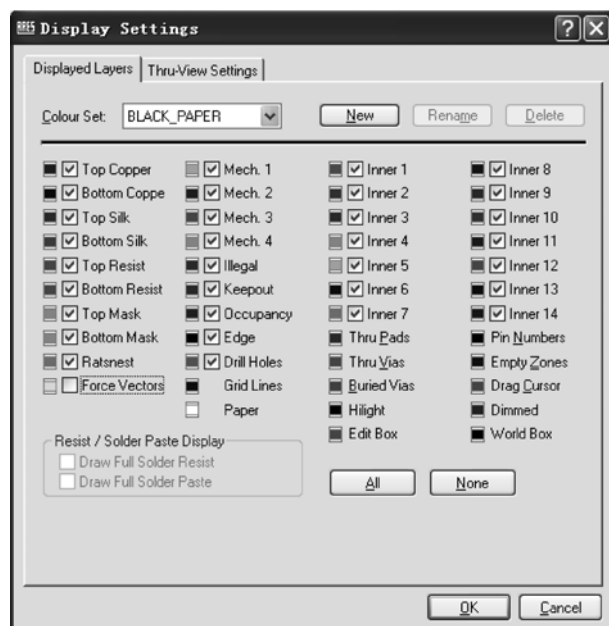


图 8-73 Display Settings 对话框

手工调整布局后，元件位置如图 8-74 所示。

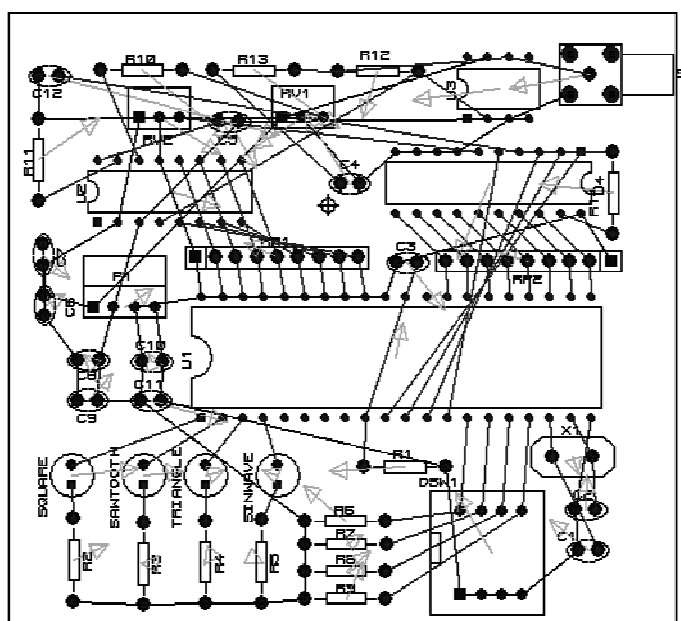


图 8-74 手工调整布局

8.3.4 调整文字

右击选中元件，在元件 ID 号上单击鼠标左键，弹出 Edit Component Id 对话框，可修改器件 ID 号、所属层面、旋转角度、高度及宽度，如图 8-75 所示。

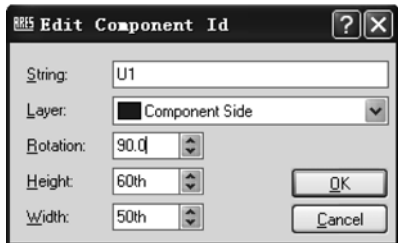


图 8-75 Edit Component Id 对话框

- ☺ String: 元件 ID 号;
- ☺ Layer: 所在层面;
- ☺ Rotation: 旋转角度;
- ☺ Height: 文字高度;
- ☺ Width: 文字宽度。

通常可以像移动元件一样移动元件 ID 号，当需要旋转时，调出 Edit Component Id 对话框，修改 Rotation 值即可。调整后如图 8-76 所示。

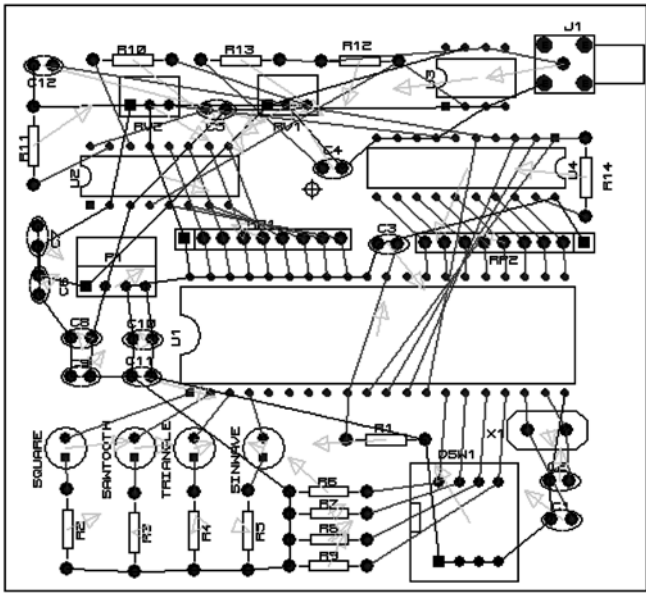


图 8-76 调整文字



8.4 PCB 布线

8.4.1 设置约束规则

在 PROTEUS ARES 界面的菜单栏中选择 Technology→Design Rule Manager，弹出 Design Rule Manager 对话框，如图 8-77 所示。



图 8-77 Design Rule Manager 对话框

- ☺ Rule Name: 规则名称。
- ☺ Apply to Layer: 应用到的层。
- ☺ Apply to Net Class: 网络种类。
- ☺ With Respect To: 与上面所相关内容。
- ☺ Clearances: 间隙。
 - ☞ Pad-Pad Clearance: 焊盘间距;
 - ☞ Pad-Trace Clearance: 焊盘与 Trace 之间的间距;
 - ☞ Trace-Trace Clearance: Trace 与 Trace 之间的间距;
 - ☞ Graphics Clearance: 图形间距;
 - ☞ Edge/Slot Clearance: 板边沿/槽间距。
- ☺ Apply Defaults: 应用默认值。

单击如图 8-77 所示对话框中的 Net Classes, 可弹出如图 8-78 所示的对话框窗口, 按照所示设置 POWER 层约束规则。

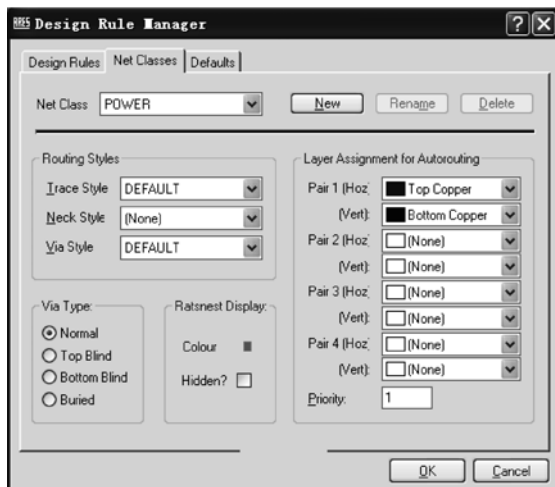


图 8-78 Net Classes 对话框

- ☺ Net Class: 网络种类, 分别为 POWER 层或 SIGNAL 层。
- ☺ Routing Styles: 布线样式。
 - ▢ Trace Style: Trace 的样式;
 - ▢ Neck Style: Neck 线的样式;
 - ▢ Via Style: 过孔的样式。
- ☺ Via Type: 过孔。
 - ▢ Normal: 普通过孔;
 - ▢ Top Blind: 顶层盲孔;
 - ▢ Bottom Blind: 底层盲孔;
 - ▢ Buried: 埋孔。
- ☺ Ratsnest Display: 构筑显示。
 - ▢ Colour: 颜色;
 - ▢ Hidden: 隐藏。
- ☺ Layer Assignment for Autorouting: 为自动布线给各层次赋值。
 - ▢ Pair 1: 层对 1, 顶层水平布线, 底层垂直布线。
- ☺ Priority: 优先级。

然后单击 Net Class 栏旁边的下拉菜单, 按照图 8-79 所示设置 SIGNAL 层约束规则。

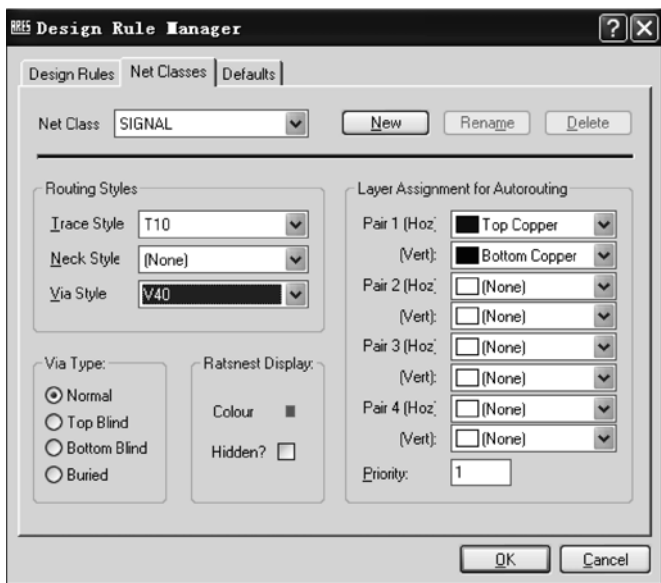



图 8-79 SIGNAL 层约束规则

8.4.2 手工布线及自动布线

1. 手工布线

用鼠标单击 PROTEUS ARES 工具箱中的 Track Mode 按钮, 在列表栏中选择线宽 T10, 左键单击一个有飞线连接的焊盘, 沿着飞线的提示开始布线, 如图 8-80 所示。

在另一个焊盘上单击鼠标左键，完成布线，如图 8-81 所示。

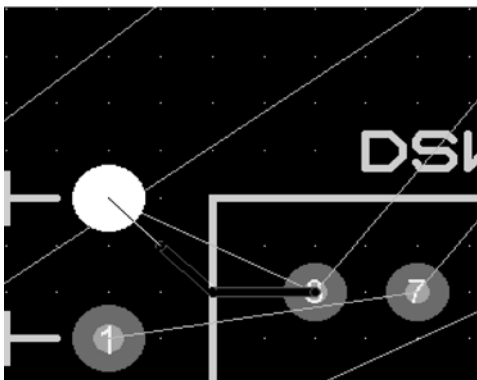


图 8-80 布线起点

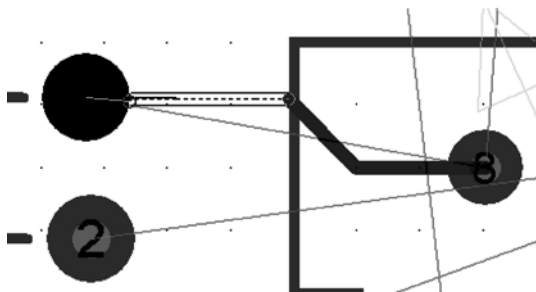


图 8-81 完成布线

注意：

电路的布线最好按照信号的流向采用全直线,需要转折时可用 45° 折线或圆弧曲线来完成,这样可以减少高频信号对外的发射和相互间的耦合。高频信号线的布线应尽可能短。要根据电路的工作频率,合理地选择信号线布线的长度,这样可以减小分布参数,降低信号的损耗。制作双面板时,在相邻的两个层面上线最好相互垂直、斜交或弯曲相交。避免相互平行,这样可以减少相互干扰和寄生耦合。

高频信号线与低频信号线要尽可能分开,必要时采取屏蔽措施,防止相互间干扰。对于接收比较弱的信号输入端,容易受到外界信号的干扰,可以利用地线做屏蔽将其包围起来或做好高频接插件的屏蔽。同一层面上应该避免平行走线,否则会引入分布参数,对电路产生影响。若无法避免,则可在两平行线之间引入一条接地的铜箔,构成隔离线。

在数字电路中,对于差分信号线,应成对地走线,尽量使它们平行、靠近一些,并且长短相差不大。

在 ARES 界面左下角的层面列表中选择布线层 Bottom Copper, 如图 8-82 所示,进行底层布线,如图 8-83 所示。



图 8-82 切换布线层

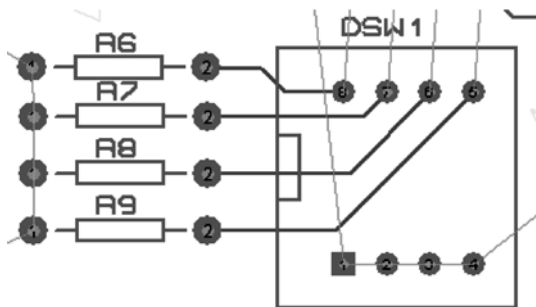


图 8-83 底层布线

在布线过程中,双击鼠标左键,添加过孔,自动切换层面,继续布线,如图 8-84 和图 8-85 所示。



图 8-84 添加过孔 1

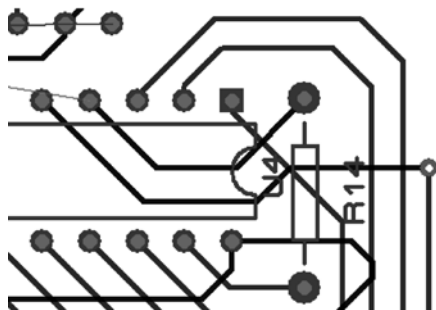


图 8-85 添加过孔 2

右键选中导线，用左键拖动，可修改连线，如图 8-86 所示。

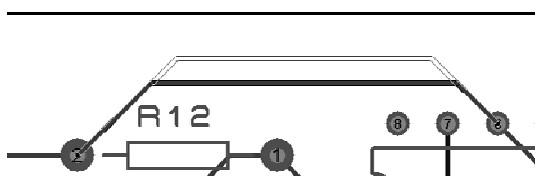


图 8-86 修改连线

2. 自动布线

在菜单栏中选择 Tools→Auto Router，弹出 Shape Based Auto Router 对话框，按照图 8-87 所示进行设置。

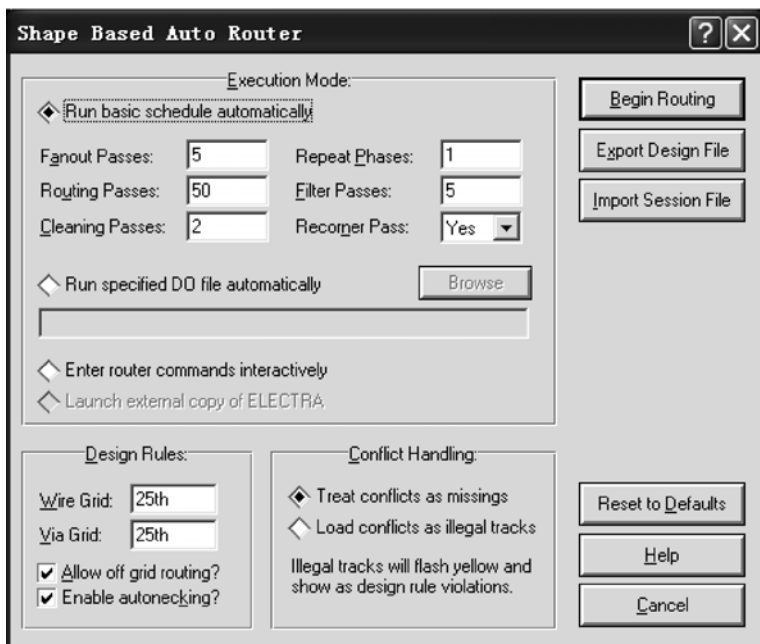


图 8-87 Shape Based Auto Router 对话框

单击对话框中的“Begin Routing”按钮，完成自动布线，如图 8-88 所示。

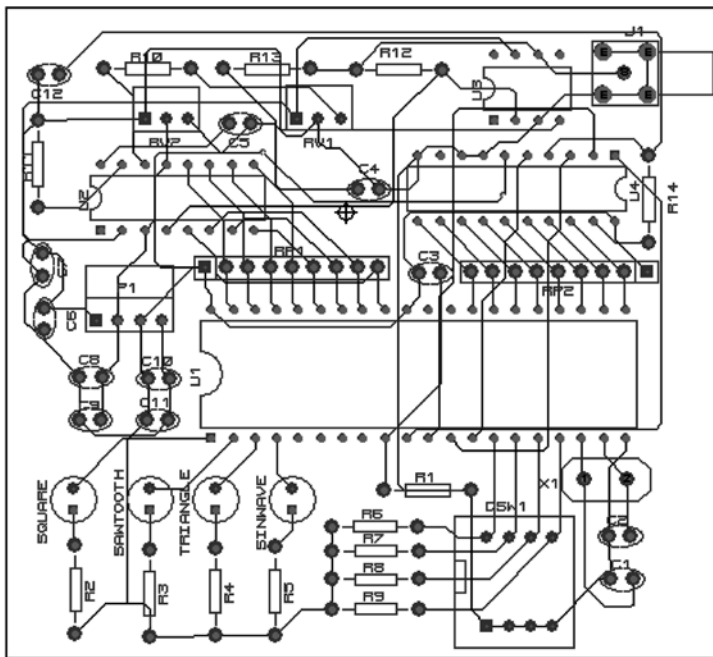



图 8-88 完成自动布线

【布线检查】

☺ CRC检查: 在菜单栏中选择Tools→Connectivity Checker, 检查多余的、遗漏的连接等情况。

☺ DRC检查: 在菜单栏中选择Tools→Design Rule Checker, 检查违反规则的物理错误。
DRC提示如图8-89所示。

布线完成后, 用 3D 效果观察器件, 单击工具栏上的 , 弹出 3D 效果图, 如图 8-90 所示。

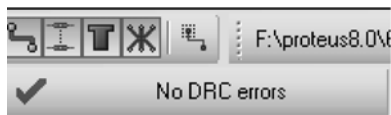


图 8-89 DRC 提示

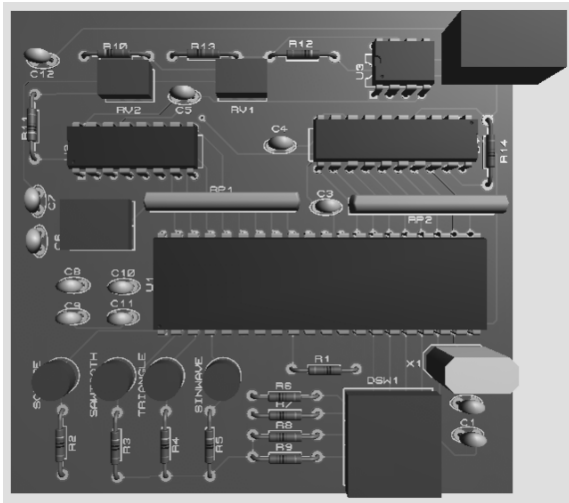


图 8-90 3D 效果图

8.4.3 铺铜

在 ARES 菜单栏中选择 Tools→Power Plane Generator，弹出 Power Plane Generator 对话框，如图 8-91 所示。

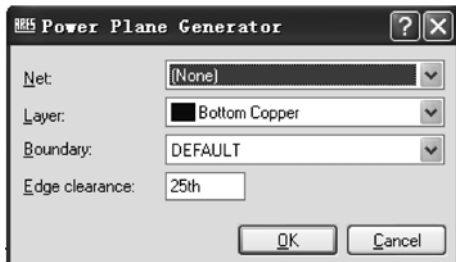


图 8-91 Power Plane Generator 对话框 1

- ☺ Net: GND=POWER，选择铺铜网络为 GND。
 - ☺ Layer: Bottom Copper，选择铺铜层面为底层。
 - ☺ Boundary: DEFAULT，使用默认的边界。
 - ☺ Edge clearance: 25th，铜皮距板边框的距离为 25th。
- 单击“OK”按钮，开始铺设底层铜皮，如图 8-92 所示。

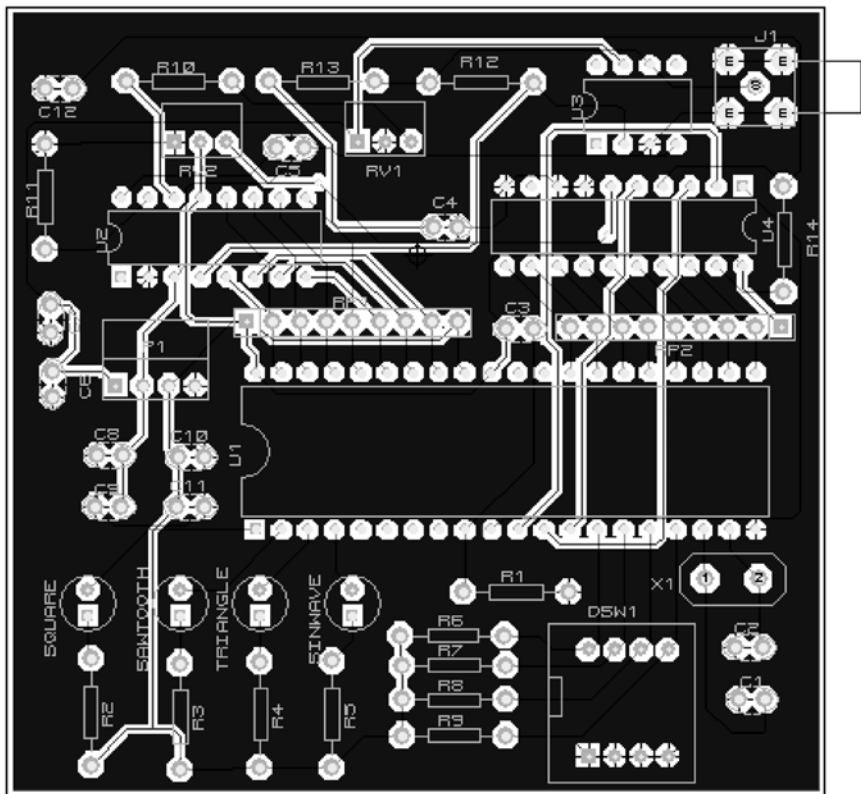


图 8-92 铺设底层铜皮

注意:

所有与网络 GND 相连的引脚或过孔都会以热风焊盘的形式与铜皮相连, 如图 8-93 所示。

打开 Power Plane Generator 对话框, 将 Layer 设置为顶层, 如图 8-94 所示。

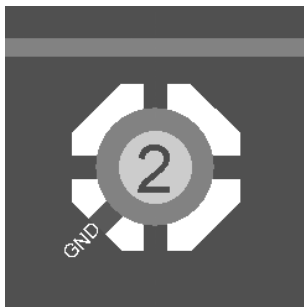


图 8-93 热风焊盘

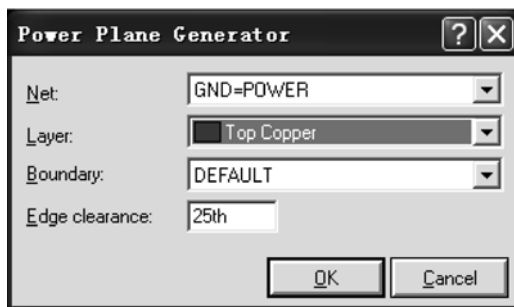


图 8-94 Power Plane Generator 对话框 2

单击“OK”按钮, 铺设顶层铜皮, 如图 8-95 所示。

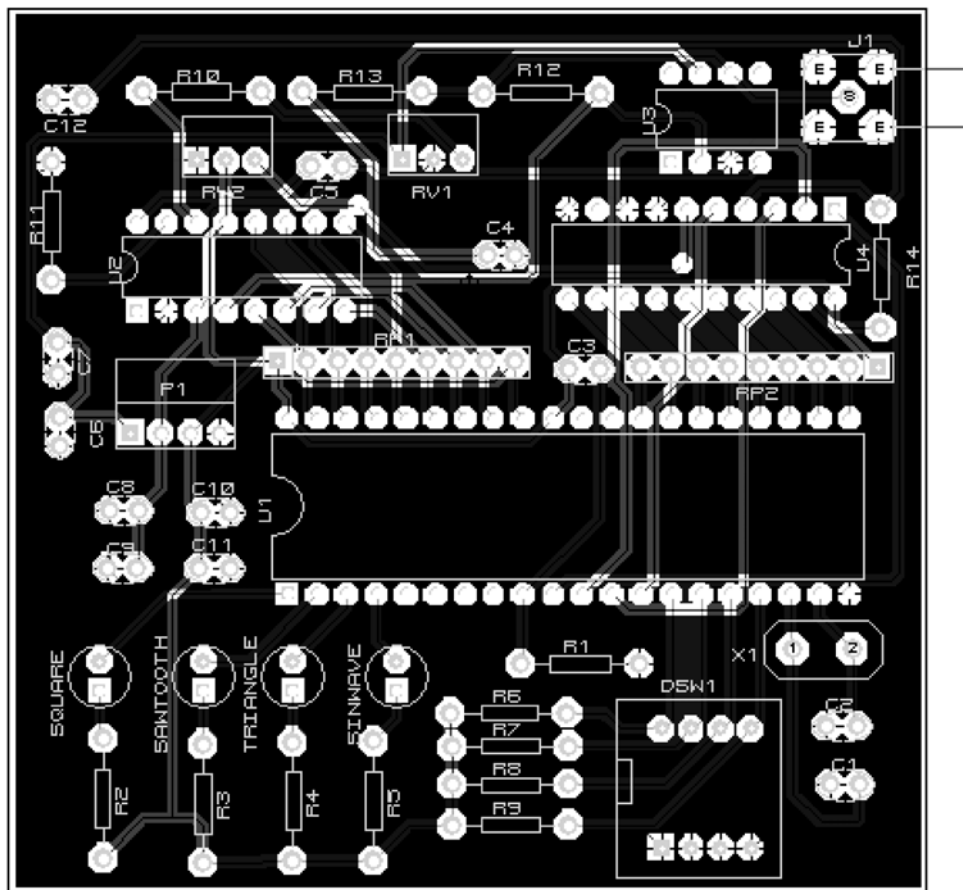


图 8-95 铺设顶层铜皮



8.5 输出光绘文件

在 PROTEUS ARES 的菜单栏中选择 Output→Gerber/Excellon Files, 打开 CAD/CAM (Gerber and Excellon) Output 对话框, 按照图 8-96 所示进行设置。

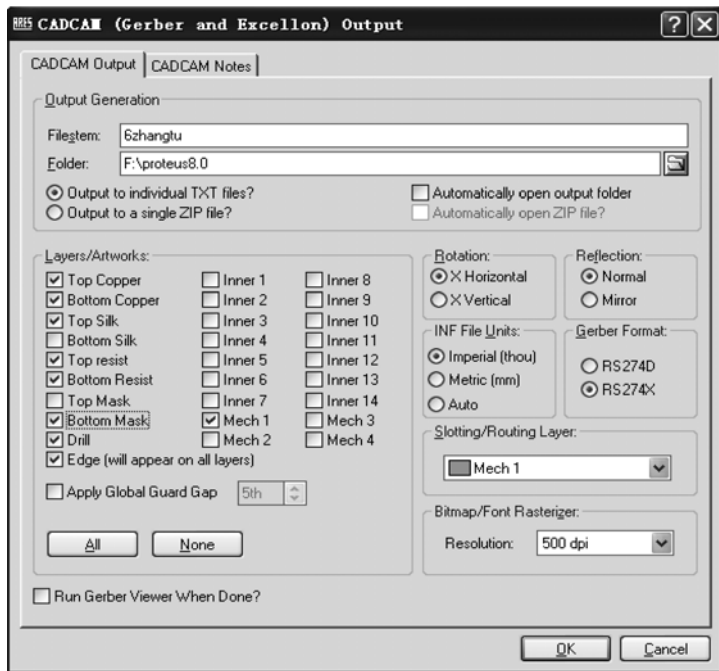


图 8-96 CAD/CAM (Gerber and Excellon) Output 对话框

选择 Run Gerber Viewer When Done 选项, 单击“OK”按钮, 生成光绘文件, 并弹出 Gerber View 对话框, 如图 8-97 所示。

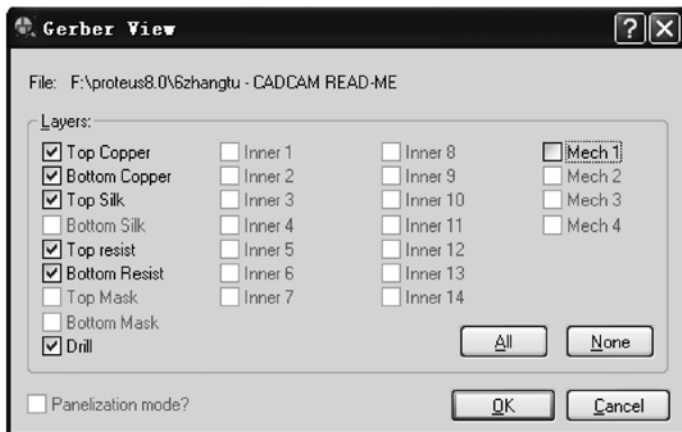


图 8-97 Gerber View 对话框

单击“OK”按钮, 查看选中的光绘层, 如图 8-98~图 8-103 所示。

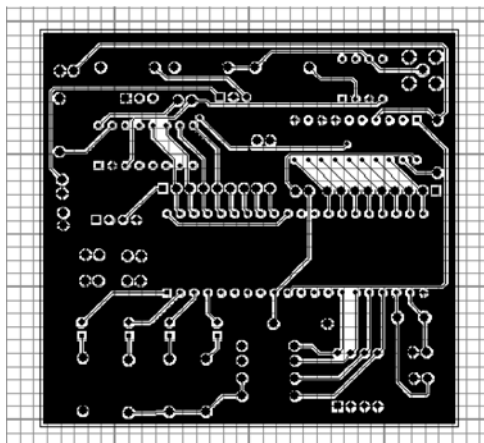


图 8-98 顶层铜 (Top Copper)

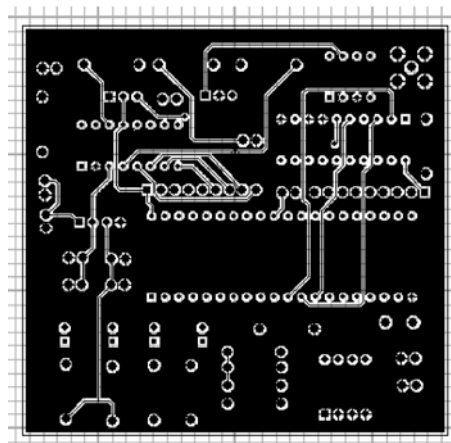


图 8-99 底层铜 (Bottom Copper)

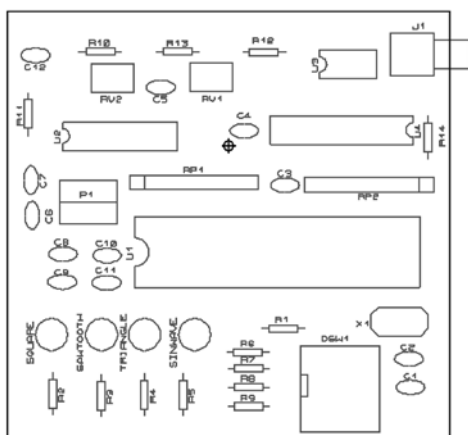


图 8-100 顶层丝印层 (Top Silk)

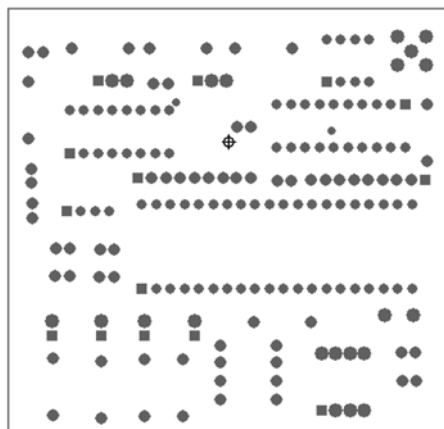


图 8-101 顶层阻焊层 (Top Resist)

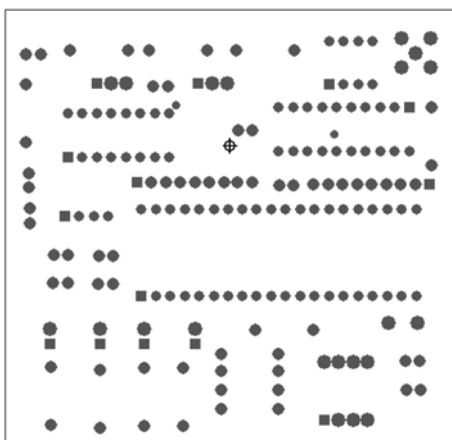


图 8-102 底层阻焊层 (Bottom Resist)

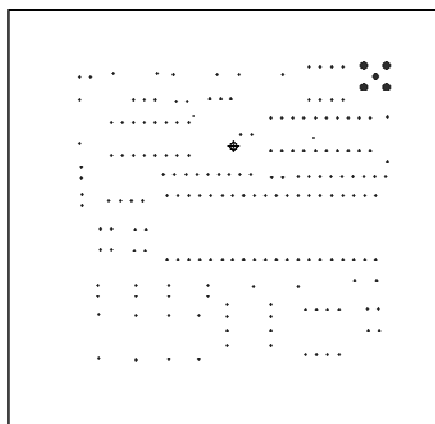


图 8-103 钻孔层 (Drill)

参 考 文 献

- [1] Labcenter 公司. PROTEUS ISIS 用户手册, 2012.
- [2] 周润景, 张丽娜. 基于 PROTEUS 的电路及单片机系统设计与仿真[M]. 北京: 北京航空航天大学出版社, 2006.
- [3] 周润景, 袁伟亭, 景晓松. PROTEUS 在 MCS-51&ARM7 系统中的应用百例. 北京: 电子工业出版社, 2006.
- [4] 周润景, 张丽娜, 丁莉. 基于 PROTEUS 的电路及单片机系统设计与仿真 (第 2 版) [M]. 北京: 北京航空航天大学出版社, 2009.
- [5] 周润景, 蔡雨恬. PROTEUS 入门实用教程 (第 2 版) [M]. 北京: 机械工业出版社, 2011.